

Up and Running With Macromedia Authorware

By Spencer R. G. Hudson

Document version: authorware-beginners.doc

Copyright © 1996-2000 Spencer R. G. Hudson
All rights reserved

You are free to copy and distribute this manual providing: (a) it remains wholly intact – free of any edits; and you (b) email is sent to me stating that you are using it and why (you must include: # of people, description of use/reason). Of course, if you have any suggestions for improvements just let me know I will see about working them into the next version.

Forward

While this manual is provide free of charge you are not free to use it for any purpose. In deed you are only permitted to use this manual for non-profit purposes (e.g. education institutions). Also, you are required to contact me and provide information as to how and why you are using the manual. In return I will contact you and let you know when the next version is available. I am currently working on a new version to take into account version 4 of Authorware. The new manual will also cover some of the more advanced techniques.

To date this manual has been used by approximately 500+ people to my knowledge. Ranging from the general public to MSc and Ph.D. students. And in the large comments have been very good. However, to remain a useful resource I require your comments.

Please let me know what you think of this manual, all comments help in the production of the next version

This short course of tutorials only scratched the surface of Authorware. There are many useful features are not been demonstrated. However, after completing the tutorials in this booklet, you should now have an appreciation of what Authorware is capable of. You should also be able to use Authorware to build quite complex applications, involving many different types of interactions.

Following on from this booklet, an extended set of notes has been created. These additional tutorials take you thought some of the more advanced facilities available within Authorware.

I hope that you *enjoy* this set of tutorials and that you don't have too many problems working through them. If you have time I would appreciate any comments on this document and the software.

Kind regards,

Spencer Hudson

spencer@ideasmerchant.com
www.IdeasMerchant.com

Contents:

| | |
|--|-----------|
| FORWARD | 2 |
| 1ST TUTORIAL - BUILDING A SIMPLE APPLICATION (OPENING SEQUENCE)..... | 4 |
| 1) <i>Opening Authorware Professional</i> | 5 |
| 2) <i>Starting a new application,</i> | 5 |
| 3) <i>After selecting a new file.</i> | 6 |
| 4) <i>Setting up the new application.</i> | 7 |
| 5) <i>Placing the “display icon” on the flow line!</i> | 8 |
| 6) <i>Using the “Motion Icon”</i> | 9 |
| 7) <i>Using the Erase Icon</i> | 12 |
| 8) <i>Using the Wait Icon.....</i> | 13 |
| 9) <i>Calculation icon - the last thing for now.....</i> | 14 |
| <i>Quick review of materials</i> | 16 |
| <i>In your own time!.....</i> | 16 |
| 2ND TUTORIAL - A PASSWORD AND IDENTIFICATION SECTION | 18 |
| 1) <i>Adding a sound.....</i> | 19 |
| 2) <i>Grouping together a section of icons</i> | 20 |
| 3) <i>Password/Identification</i> | 22 |
| 4) <i>More elegant interactions.....</i> | 28 |
| <i>In your own time:.....</i> | 34 |
| <i>Summary</i> | 35 |
| 3RD TUTORIAL - BUILDING A SIMPLE GAME/PUZZLE..... | 36 |
| <i>In your own time.....</i> | 44 |
| <i>Summary</i> | 44 |
| 4TH TUTORIAL - SCORES AND HYPERTEXT HELP SYSTEM | 45 |
| 1) <i>More elegant interactions - timer facility.....</i> | 45 |
| 2) <i>A Hypertext Help System</i> | 46 |
| <i>Summary</i> | 56 |
| <i>In your own time.....</i> | 56 |
| 5TH TUTORIAL - ADVANCED FEATURES INVESTIGATION..... | 57 |
| 1) <i>Towards more intelligent interactions, a questionnaire!</i> | 57 |
| <i>Summary</i> | 72 |
| <i>In your own time.....</i> | 72 |
| 6TH TUTORIAL - SUMMARY “WHAT NOW?” | 73 |
| 1) <i>Variables/Functions.....</i> | 73 |
| 2) <i>Defining tabs.....</i> | 76 |
| 3) <i>Defining fonts.....</i> | 78 |
| 4) <i>Defining buttons</i> | 79 |
| <i>Summary</i> | 82 |
| <i>In your own time.....</i> | 82 |

“Up and Running With Macromedia Authorware”

This set of introductory tutorials to Authorware version 3.x will have a common theme running all thought them. We will be creating a simple puzzle. The tutorials will cover the creation of an (1) opening sequence; (2) a password and user identification section; (3) puzzle; (4) scores and help system, embedding hypertext; (5) advanced features - building a questionnaire and saving to a file; (6) summary...

1st Tutorial - Building a simple application (opening sequence)

In this tutorial some of the basic features of Authorware professional will be shown. We will be creating the opening sequence to the puzzle application. In this tutorial we will be investigating the use of the following icons.

- Display Icon
- Wait Icon
- Erase Icon
- Motion Icon
- Calculation Icon

We will also be examining how an application is set up, including:

- Default screen size
- Default background colours

A brief introduction to some other Authorware features will also be given:

- Importing images

Note: The operating system used to create these notes was Windows 95. Therefore screens may differ in some ways to those that you experience. If you do have any difficulties about this or any other matter please ask for help. This document is intended to be used in conjunction with a very *hands on* and *interactive* tuition style - and without your active participation things will be harder.

1) Opening Authorware Professional

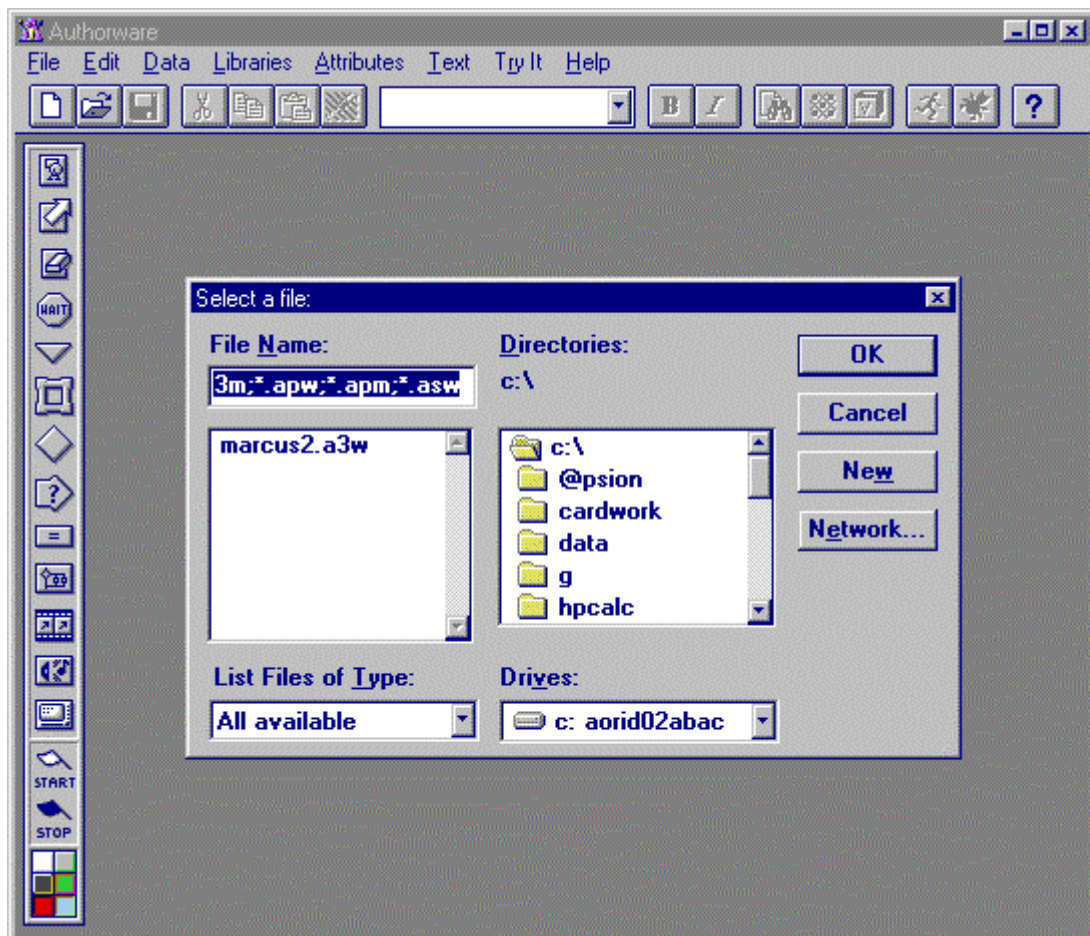
Click on the Authorware professional icon.



The Authorware 3.x icon

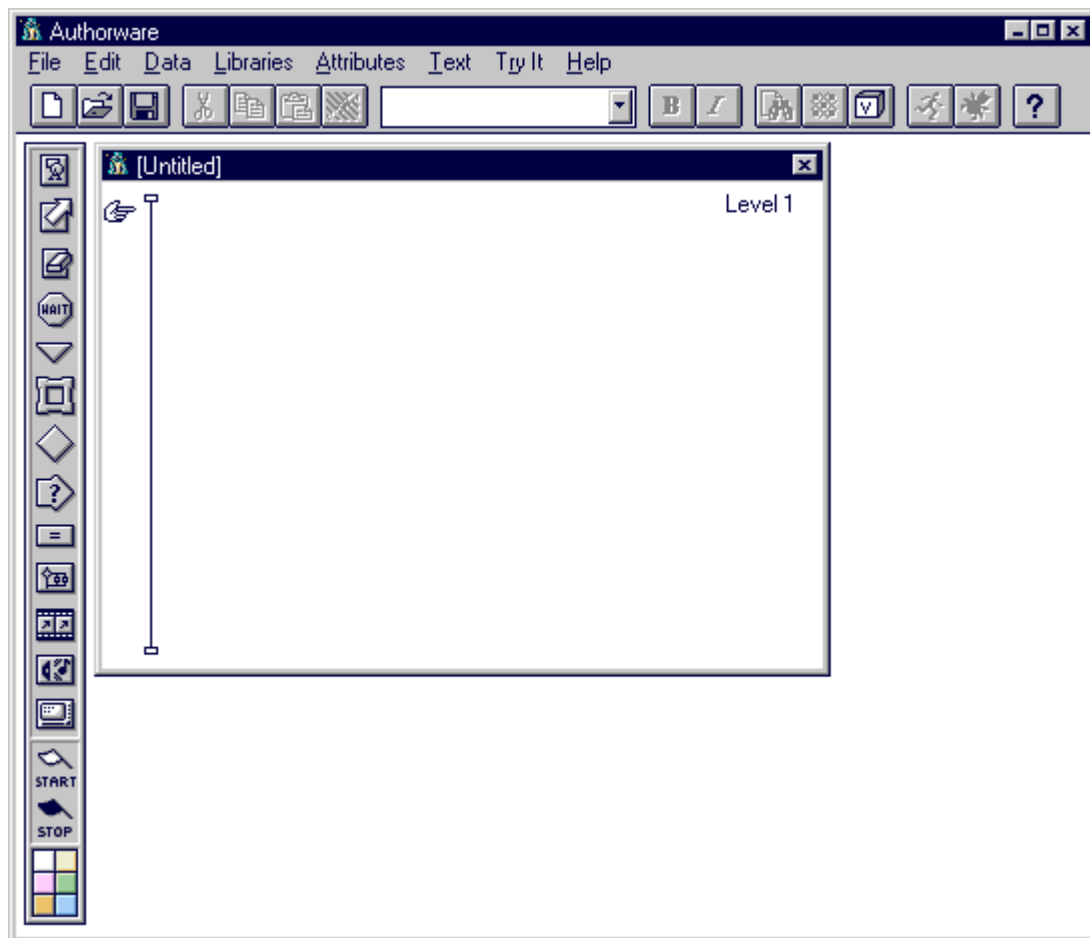
2) Starting a new application,

Select “new” from the dialogue box. Note that the opening message box has not been shown.



3) After selecting a new file.

Notice the flow line is empty (but not for long)!



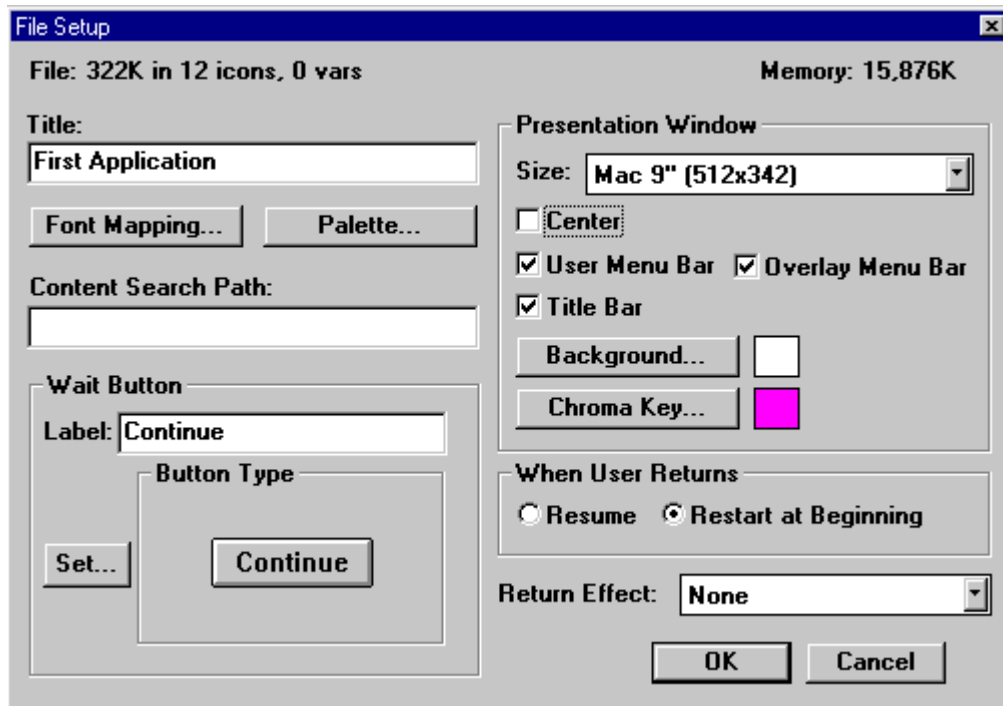
Help is always available within Authorware. Try accessing it from the “Help” menu. Before asking questions try using the help facility, and searching on keywords!

4) Setting up the new application.

Before we can begin programming the application we need to set up some of the applications attributes (i.e. screen size). These options are set by opening the “File” menu, followed by the “File setup...” option. The following dialogue box is shown.

Do the following:

- Provide the application with a title, e.g. “First Application”
- Select the “Size” option a and select “MAC 9 (512x342)”

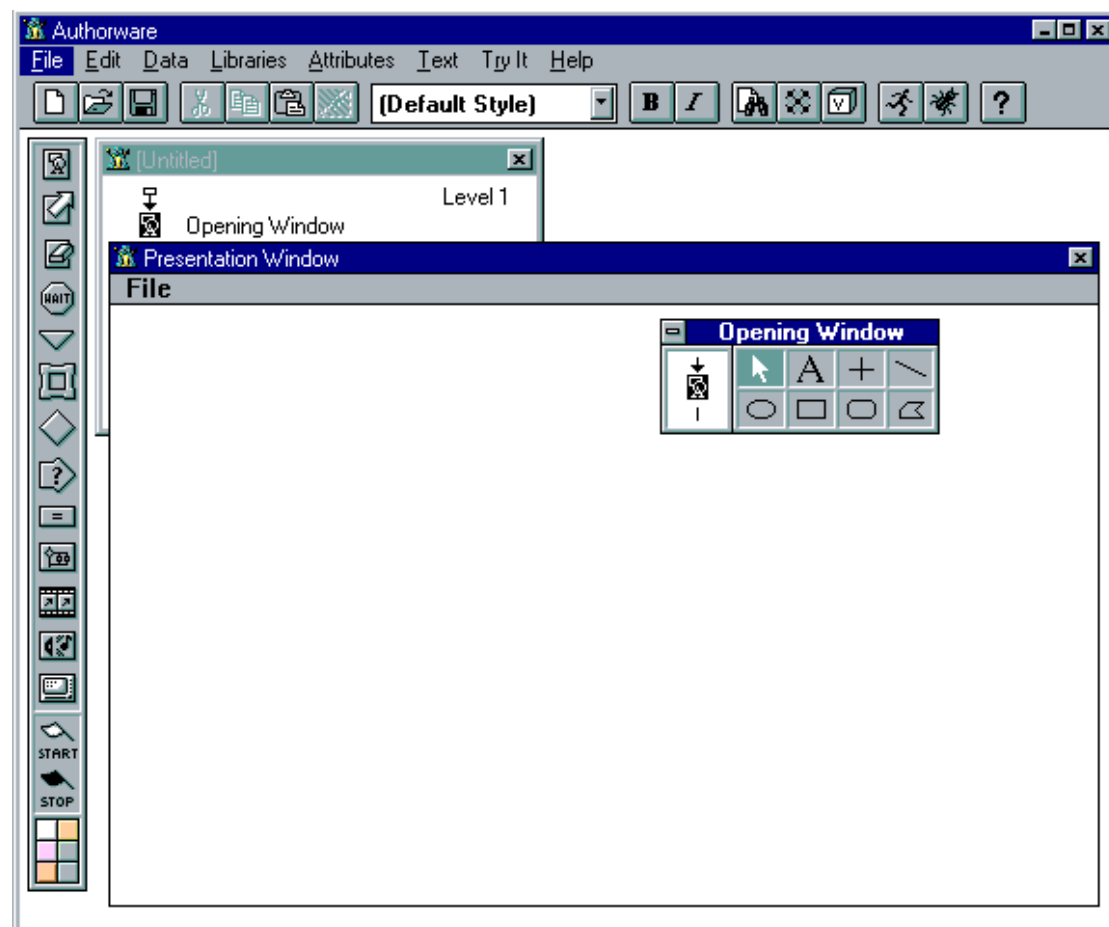


5) Placing the “display icon” on the flow line!



The display icon

- Click on, and drag the display icon onto the flow line. A more complete flow line is shown on the right.
- Change the “untitled” label to something different (e.g. opening title page)
- To edit the contents of the “display icon” just double click on it. Below is an example of what you will see. Notice the little editing tools dialogue box that becomes available. This is always opened when a display icon is opened.



You will now be able to place text, lines, circles, boxes etc. on the surface of the display icon. Of course you will want to change the default attributes of elements (e.g. colour, font, line size and so on). These are changed through the menus - which can bring up more tools from under both the Attributes menu (i.e. Effects..., Lines..., Fills..., Modes..., Colours...; Bring to front, Send to back; and so on) and the Text menu (i.e. Font, Size, Style, Alignment, Scrolling, ...).

Enter into the display icon your name and course. Also add a coloured back ground! These features are explained at the end of this section.

6) Using the “Motion Icon”

Next we will move an object around the screen. However, before we can do this we will need an object to move. For this reason add another display icon onto the flow line and place a ball in it - it is the ball that we will be moving around the screen.

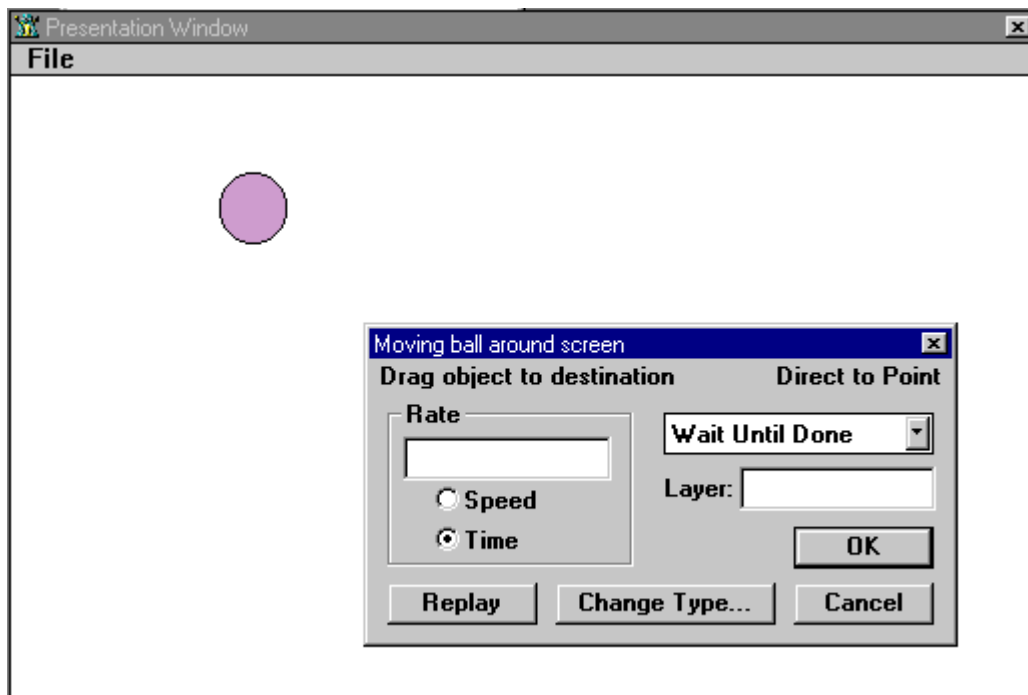
The flowline should now look something like the one on the left. Notice that I have been labelling the icons as I go along. The importance of this will become more obvious as applications become more complicated.



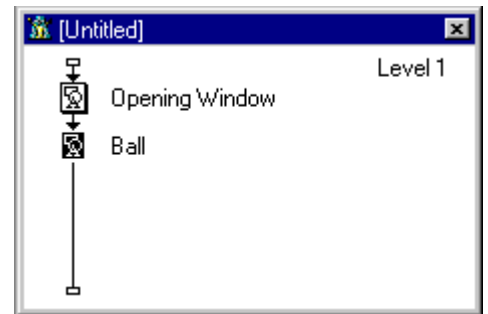
Motion Icon

We are now ready to move the ball around the screen.

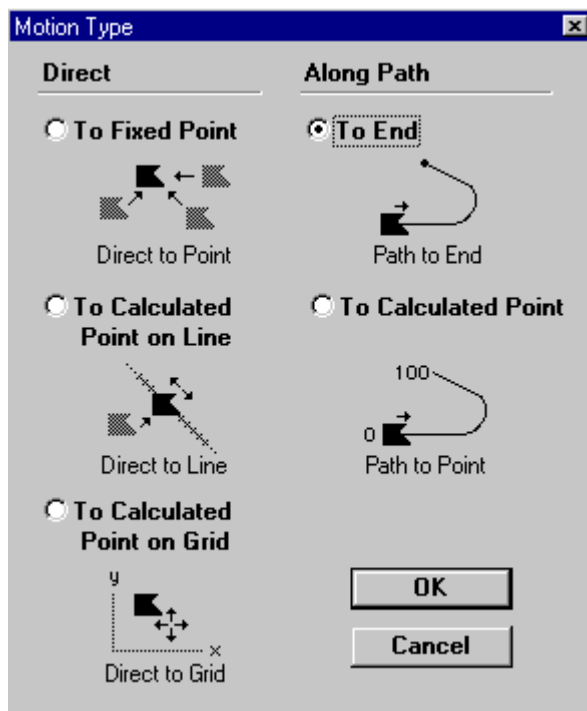
- Click on and drag the motion icon onto the flow line (i.e. below the ball display icon you have just added, as instructed in the above paragraph).
- Label the motion icon “Moving ball around screen”
- Double click on the ball display icon and close the window (this tells the software what element it is that we want to move).
- Next double click on the motion icon.
- You should see something like this:



- Click on “change type”



- You should now see the following dialogue box.



- Now select “To End” as the type of motion that we want to see.

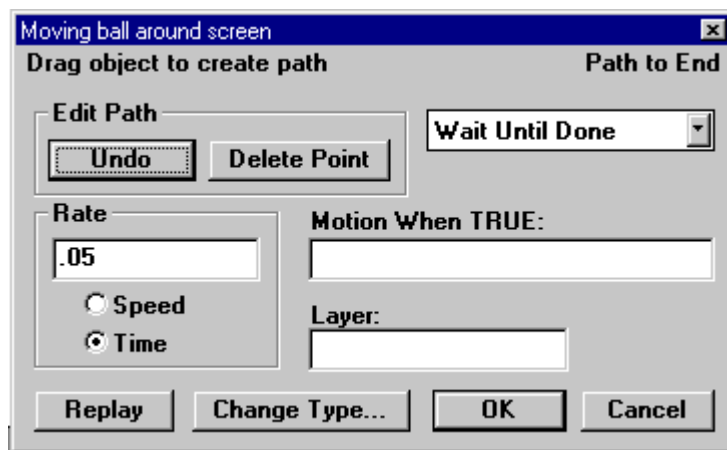
Click on the “OK” button - please note that simple steps such as this will be assumed and not always detailed in the instructions - use your common sense.

Learning is better achieved when you explore - if you do get stuck ask a friend or the tutor.

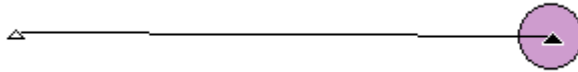
- Next click on the circle, you will notice a little triangle appear within the ball. This denotes both the starting point of the motion and the type of path (i.e. straight) it will be taking.



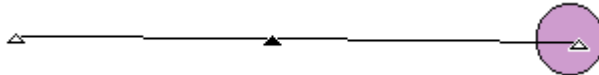
- You will also notice that the dialogue box has change to take into account the new type of motion.



- Next click on the circle and move it across the screen.



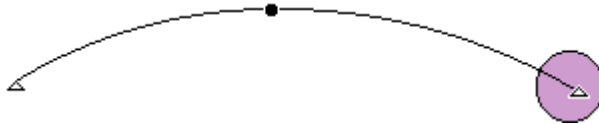
- You will notice a line is drawn - this indicates the path the ball is to take. Also, you will also notice that at the beginning and end of the line are little triangles - more on this later.
- Now press replay. You should see the ball move from the starting point to the end point. Try playing with the Rate settings (enter any number and select time then speed to see what happens - to begin with select 5 and time).
- Now we will add another point in the line.
- Click on the middle of the line. You should now see three triangles!



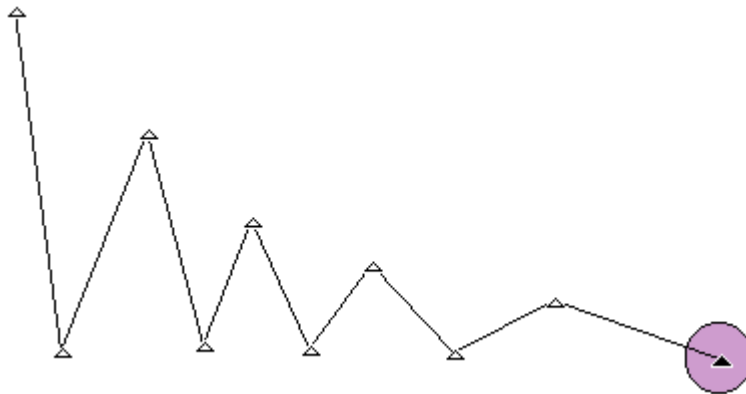
- Next, click and drag the middle triangle up.



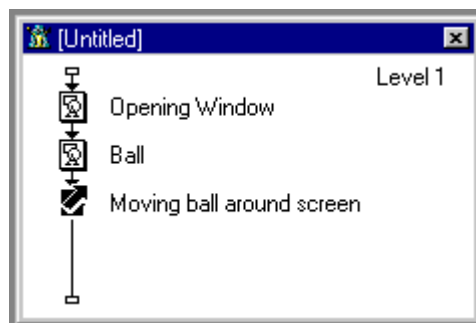
- Click on replay and see the new path taken.
- Now double click on the middle triangle and notice it change into a circle, also notice the shape of the path has changed (i.e. circles denote a curved path and triangles a straight path)



- As many mid-points can be added to a path as you like. Try the following example and make the ball bounce.



- You should now have a flow line that looks something like this



- We can now run the application to see what it is look like. Do run an application go to the menu and select “Try It” and then run. Notice that you could also press CTRL+R.

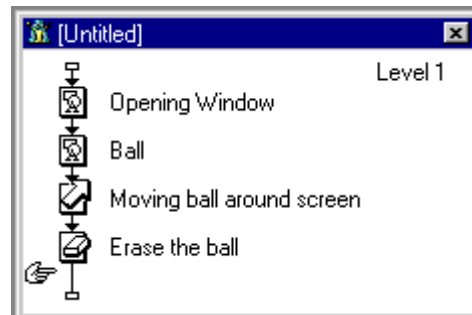
7) Using the Erase Icon

Now that we can put elements on the screen, and move them around next we need to be able to erase them. To do this we use the erase icon

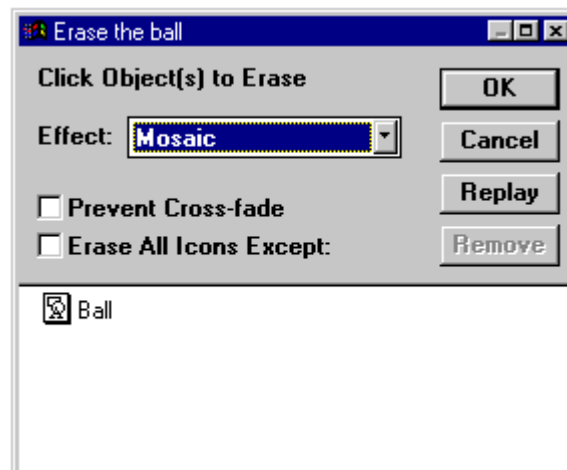


Erase Icon

- The erase icon is very easy to use. Just add it to the flow line in the usual way. (Don't forget to give the icon a description, e.g. "erase the ball")



- Next, open and close the icon that we want to erase. This shows Authorware which elements we want to erase
- Now double click on the erase icon
- You will now see the usual application window with the ball in it
- Click on the ball and it will be added to erase dialogue box



- Notice that I have selected an erase effect
- Have a look and see what other erase effects are available and what they do
- Run the application now (CTRL+R) and see how things look

8) Using the Wait Icon

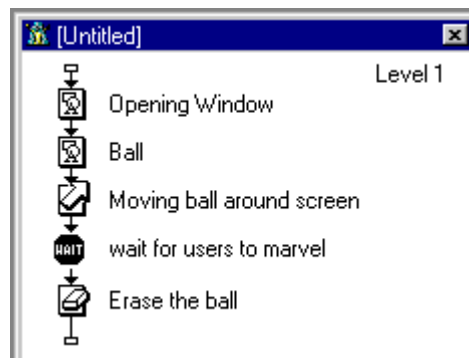
You may have noticed when you last ran the application that as soon as the animation had finished the ball was erased. There is a facility in Authorware to **wait** a defined period before continuing. In this case it would allow end-users to marvel at your brilliant motion of the ball.

To make Authorware wait we use (unsurprisingly) the wait icon

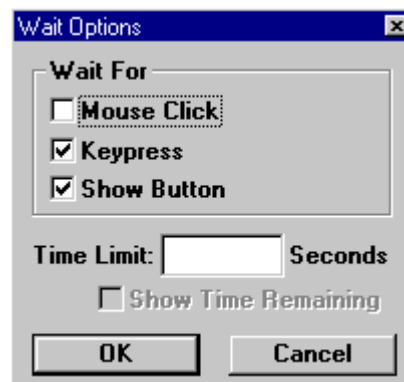


Wait Icon

- We will put the wait icon between the motion and erase icons.



- Now double click on the wait icon and define the wait properties



- Within the wait icon there are a number of options that can be selected. We can make the end-user wait until: (1) they press a mouse; (2) make a key press; (3) they press a button. We can also define a time period. If we do this we can also put select to put a little clock on the screen that counts down the time
- Try running the application now and see what happens

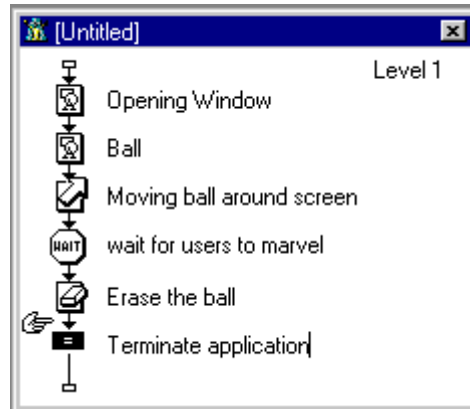
9) Calculation icon - the last thing for now

When running the application the last time you may or may not have noticed that at the end nothing happens. The screen just stays there. I will now explain how to quit your application properly. To do this we will need to add one final icon to the program.

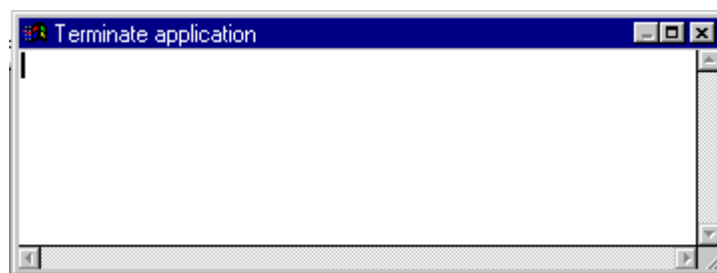


The calculation icon

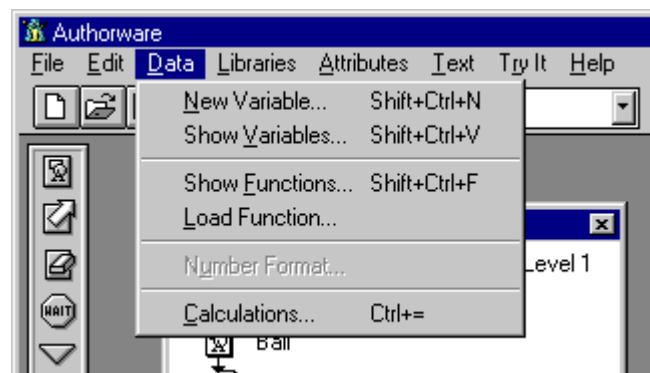
- Drag a calculation icon to the end of the flow line, (give it a suitable name).



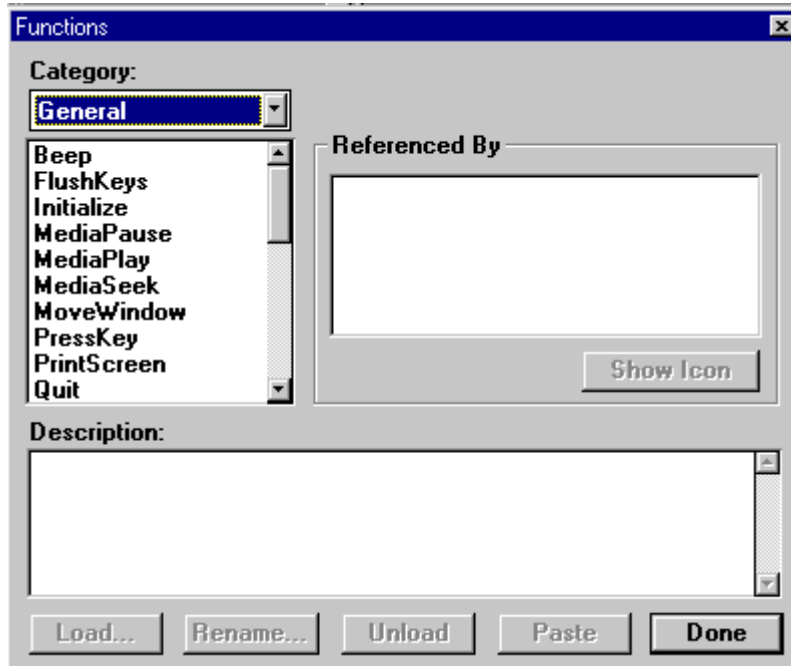
- Now double click on the icon, you will be greeted by a text entry area.



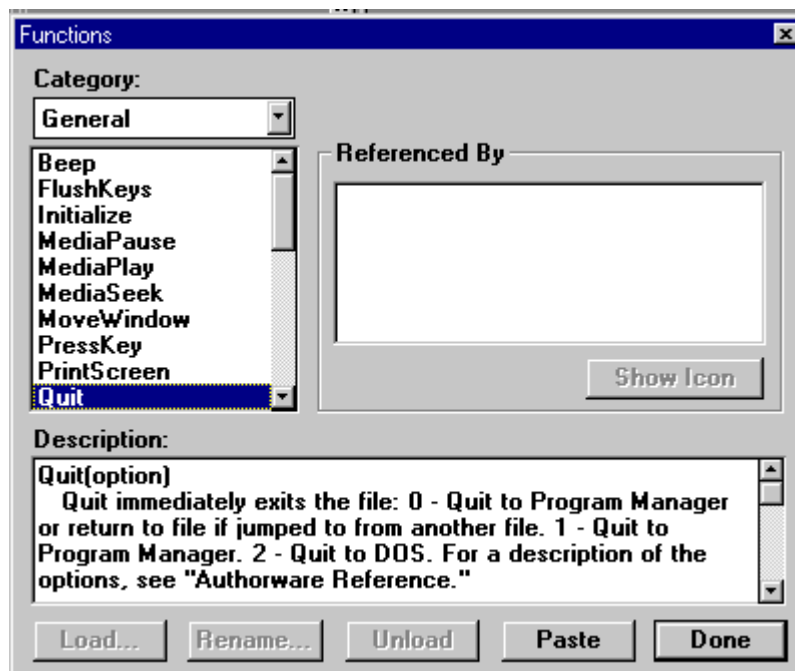
- As I can't remember what the command to quit an application I will take you through the online assistance feature.
- First, go to the menu and select Data, then Show functions, (notice the keyboard short cut SHIFT+CTRL+F)



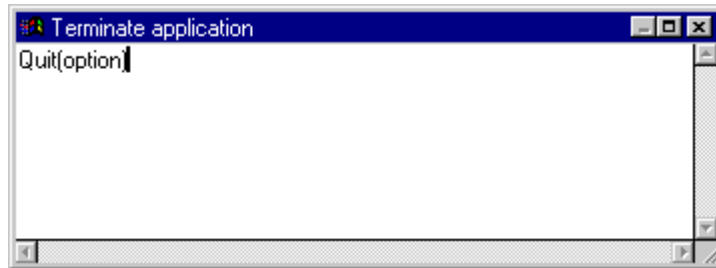
- You are now greeted with a list of the different types of functions (programming commands) available within Authorware. (I suggest that you spend some time here and investigate the different functions available).



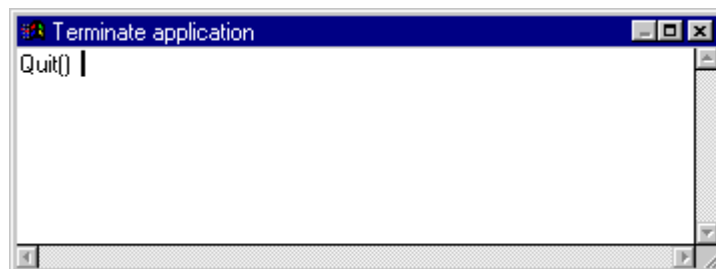
- We want to look for an exit command, and as it happens it is on the bottom of the first list. Show below is the information that is presented when you click on this function.



- At this point Authorware does something really quite nice. It assumes that as you wanted to look up the command you will want to use it. And rather than having to write down the function on a piece of paper to then put it into the calculation icon we have the “paste” button.
- Click on the paste button, then the done button



- The calculation icon has now had **Quit(option)** put into it, the command to terminate the application.
- Delete the word option from between the brackets, then close the dialogue box. It is important not to delete the brackets, just remove the word **option** from between them, see below:



- Run the application now (CTRL+R).
- Notice the difference?

Quick review of materials

- Run application = CTRL+R
- Show functions = SHIFT+CTRL+F
- Quit application = “quit()”
- Motion paths: (1) triangle = straight line; (2) circles = curved lines.
- Erase icon - for quickness use ‘erase all except’ option
- When the application is running press CTRL+P to pause and edit the objects on the screen!

In your own time!

Setting up an application:

- The “Button Type” dialogue option. Within this (i.e. click on Set) you can change the look of and interactions of default buttons within Authorware. This includes attaching sounds to buttons when they are pressed - or even when the mouse goes over the button!
- These settings can be reviewed/changed at any time.
- The reason for configuring the screen size to the small 512x342 is so that you can see both the screen being developed and the authoring tools. This will become more obvious as time progresses.

Erase Icon:

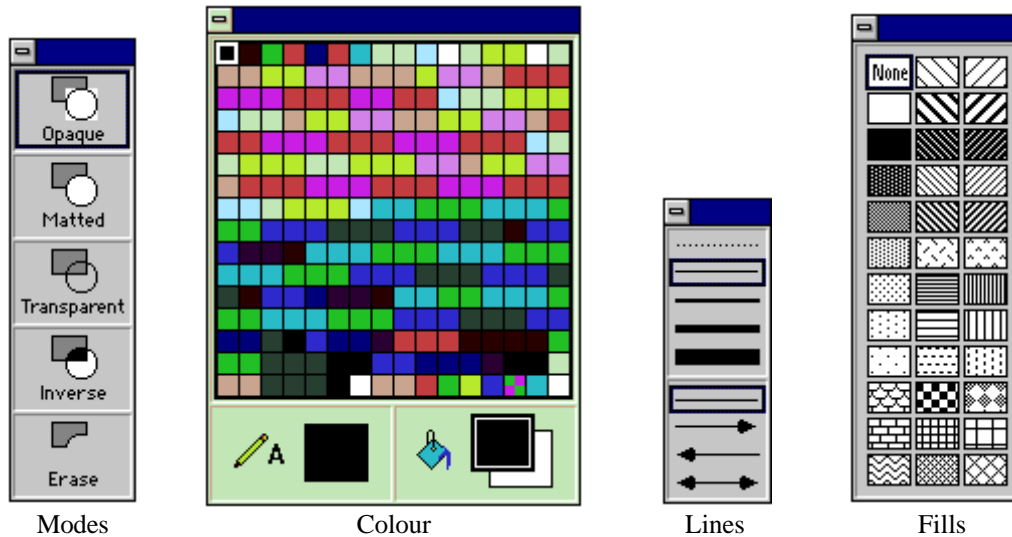
- Sometimes there are a great deal of elements on the screen and selecting the “erase all icons except” (those in the erase dialogue box) is a faster option.
- As many elements as you like can be added to the list to be erased (or not erased). A scroll bar is shown when there are more than can fit in the small space!

Wait Icon

- The continue button and the clock counting down time left can both be moved on the screen. This is done by pausing the application while running (CTRL+P), making the alterations and continuing (i.e. press CTRL+P again).
- It is considered good interaction practice if when using a wait icon a message is placed on the screen informing the users of what is happening, (i.e. either show clock or show button).

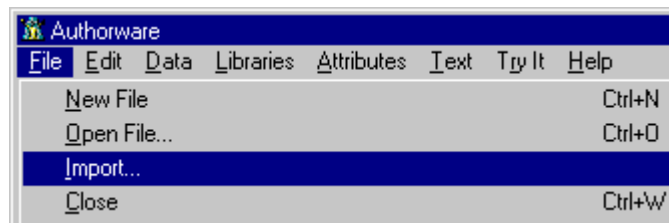
In General

- Use the Sound Icon - it is very similar in operation to those that have already been described!
- Other features of particular interest - it is strongly suggested that you should explore how these features can be used.

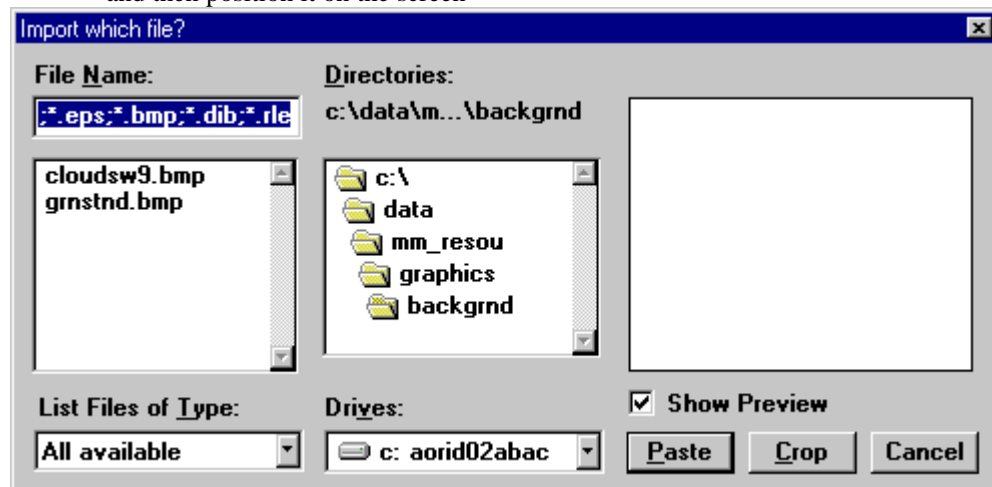


Images can be imported and shown in Authorware via the Display Icon by a number of different mechanisms. (1) you can use the cut and paste facility offered by the operating system; or (2) when we are editing a Display Icon we can import a file. The following instructions show how to import an image.

- 1) Open the Display icon
- 2) Select File, then Import... from the menu (no quick key options)



You will now see the importing dialogue box. Simply select the image you wish to import and then position it on the screen

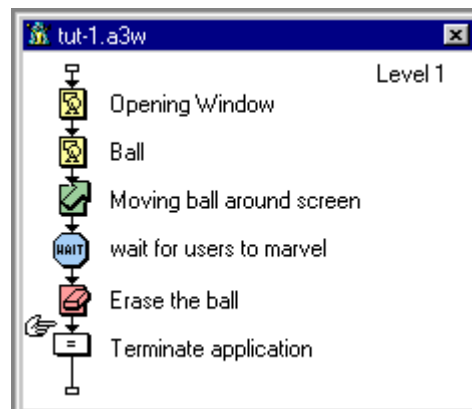


2nd Tutorial - A Password and Identification Section

In this tutorial you will be introduced to the interaction icon. You will also build upon your experience with the display icon.

In the previous tutorial we explored the use of the display icon, the motion icon and the wait icon. You were also asked to have a look at the sound icon. The introduction was intended to allow you to create an opening screen to the application that we will be creating.

Below is one example of an opening sequence. In this opening sequence a sound is played, a background image is placed on the screen, then a ball and the opening message. Finally the ball is moved around the screen. We will now add to this the part of the application - the password and identification section.



In this next section we will be making use of the following icons:

- Sound Icon (review)
- Group Icon
- Interaction Icon (three types: text, tries limit and the time limit)

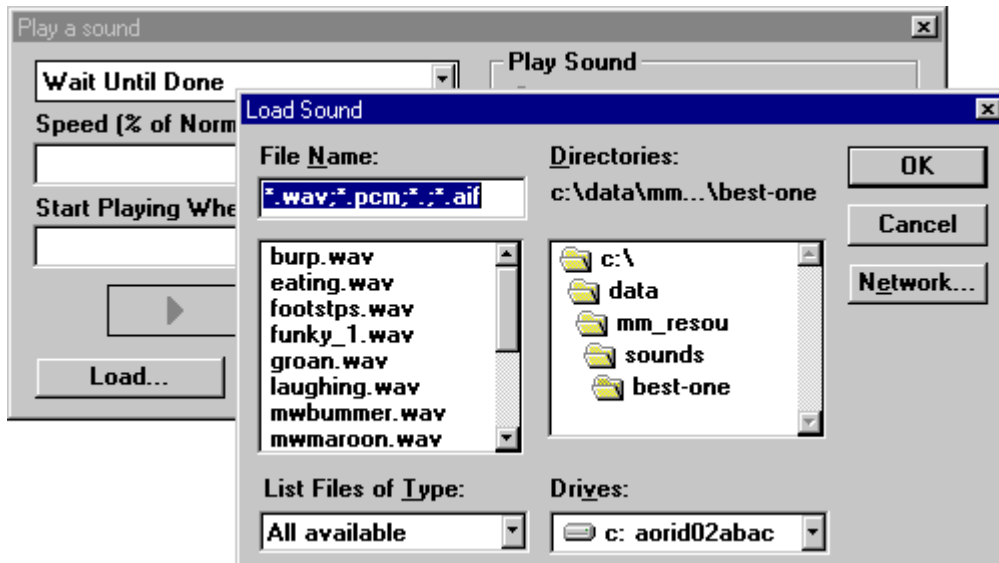
1) Adding a sound

For those of you who have not played with the sound icon, it will now be quickly reviewed. The sound icon is used to add a sound to the application.

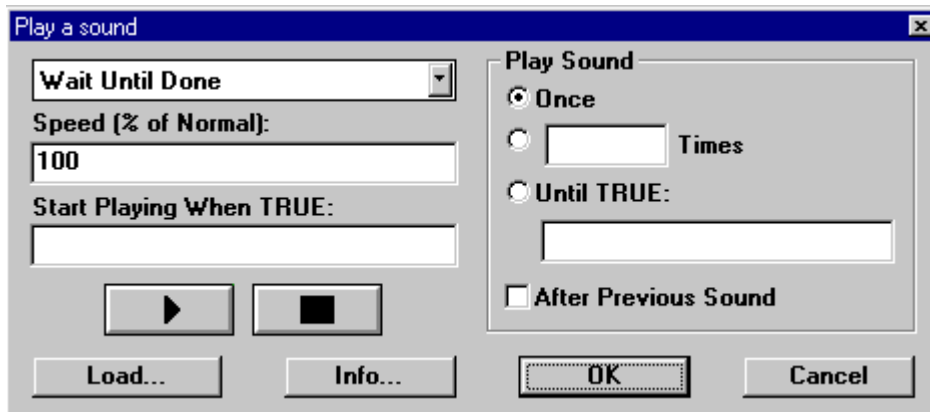


The Sound Icon

- Drag the sound icon to the flow line
- Double click on the sound icon, you will not see the following dialogue box:



- You now have to select a sound file to be played, select any.
- Then you will be shown the main sound icon dialogue box:



- The main option to consider at this moment is the drop down box that says “Wait Until Done”. This says that the application will wait until the sound file is played before continuing onto the next icon on the flow line. It is suggested that you change this option to “concurrent”.
- If you decided to load a different sound file select the “Load” button.
- Note only wave files can be played with the sound icon. MIDI files are accessed a different way.

So now everyone should be at the same point!

2) Grouping together a section of icons

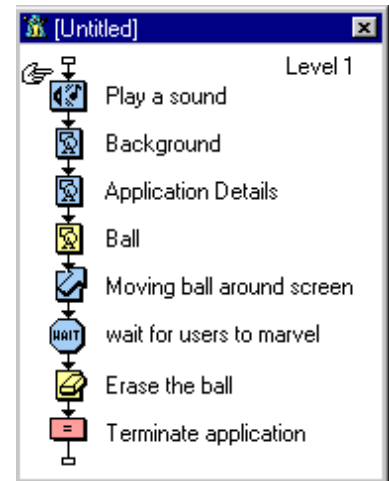
Before we begin using the Interaction icon it is useful/essential to know how to group icons together. We will practice this operation on the introduction set of icons.



The group icon

On the screen your application should now look something like the picture on the left. I have made a few changes to the application and you should update yours to.

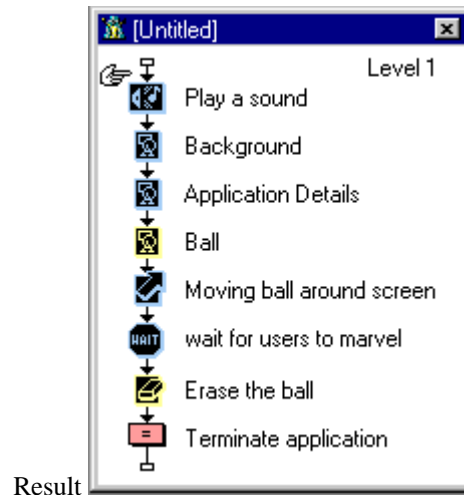
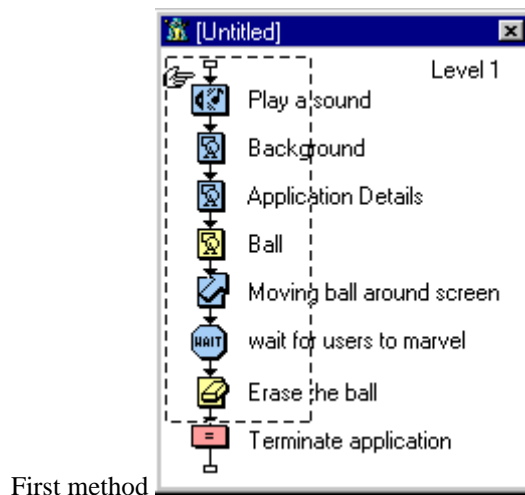
- I have imported an image and placed it in the Background display icon
- The application details display icon just contains *details* no graphics!
- These elements have been separated so that I can do individual erases on the elements. Also so that I can leave the background visible for the whole application (i.e. never erase it)



We are now going to group together all the icons (except the terminate icon) .

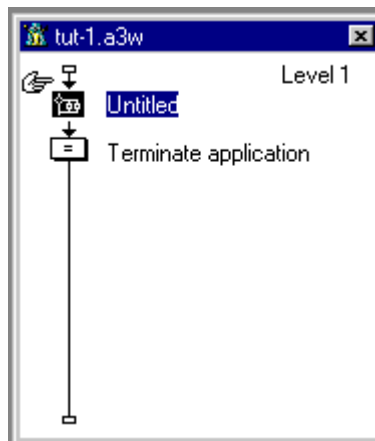
- First, highlight/select all the icons, this can be done in a number of ways. The first one to be demonstrated is the “mouse box”. This is where you click at the top left and hold the left button down, then drag the box so that it covers all the icons you want.

The second method and possibly easier one is to click on each icon in turn whilst holding down the shift button. Both methods result in the icons being highlighted.

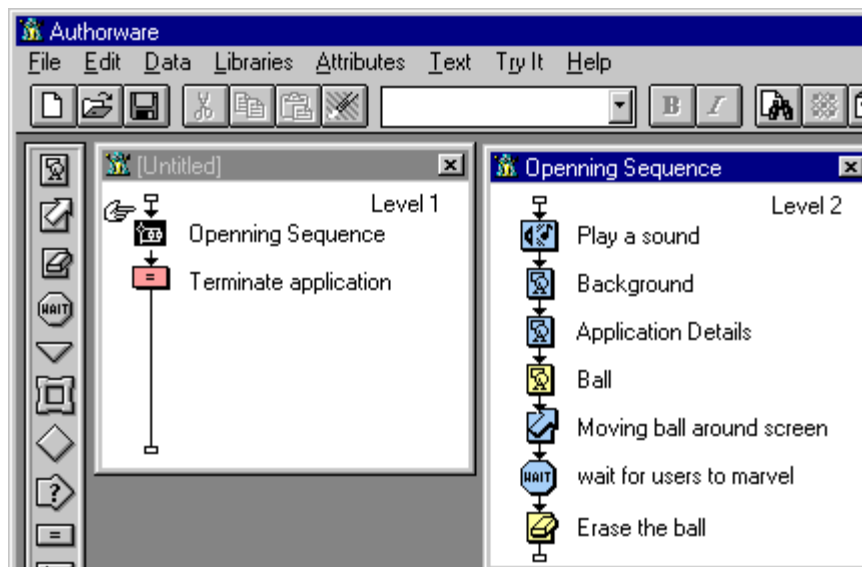


- Now press CTRL+G, this tells the computer to group together all the selected icons into one.

- You should now see the following:



- Give the Group icon a title, e.g. “Opening Sequence”.
- If you want to re-examine the icons that have *disappeared* into the group icon just double click on it, You will then see some think like:



- Notice how the flow line is now a lot less complicated. Also how the flow line is “Level 1” and the group icon is “Level 2”.
- The Authorware windows (i.e. “tut-1.a3w” and “Opening sequence”) can be moved around and sized on the screen as you wish.
- As well as grouping icons together you can also drag a group icon to the flow line and then *fill it*.

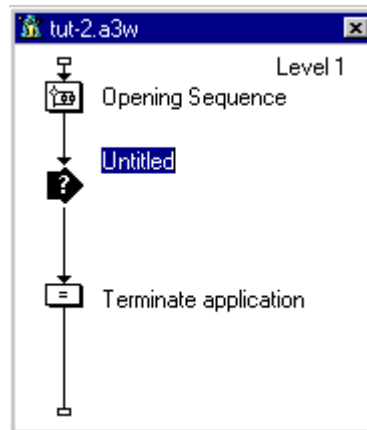
3) Password/Identification

You are all familiar with a password system and that is what we will be building next, a simple password system. To do this we will make use of the interaction icon.



The Interaction Icon

- Drag an interaction icon to the flow line:



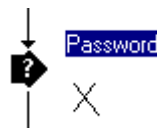
- Don't forget to give it a title (e.g. "password")
- With our password system there will be three options: (1) get it right; (2) get it wrong; and (3) try again.

If the end-user uses the correct password then they will progress onto the next part of the application.

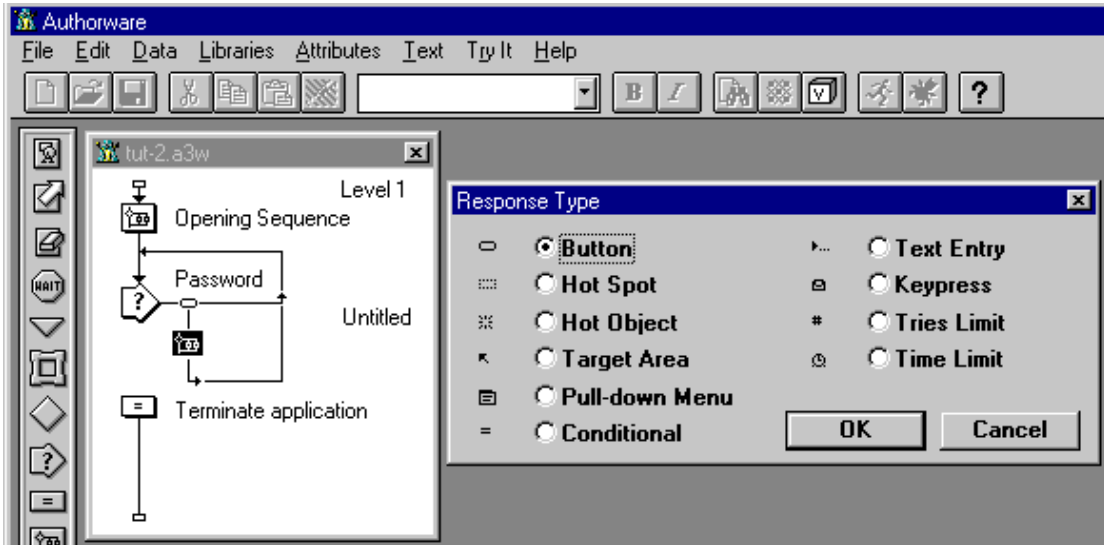
If they get it wrong they will be allowed two more attempts.

If they fail to enter the correct password an error message will be displayed and then the program will exit.

- First we will deal with the correct password!
- Drag a group icon to the right of the interaction icon (X marks the spot!):



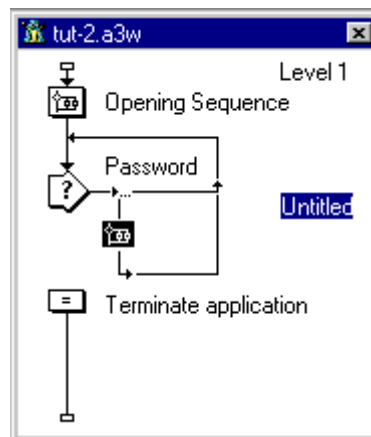
- You will now see something like this:



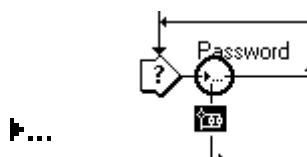
- We now have to tell the computer what type of an interaction we would like to have. Of course, we want to get a password, therefore it is a “Text Entry”.
- Select the “Text Entry” option
- Notice the little icon that appear above the group icon and to the left of the “Text Entry” option in the dialogue box above. These are the quick reference icons - Authorware is a very visually orientated development environment.



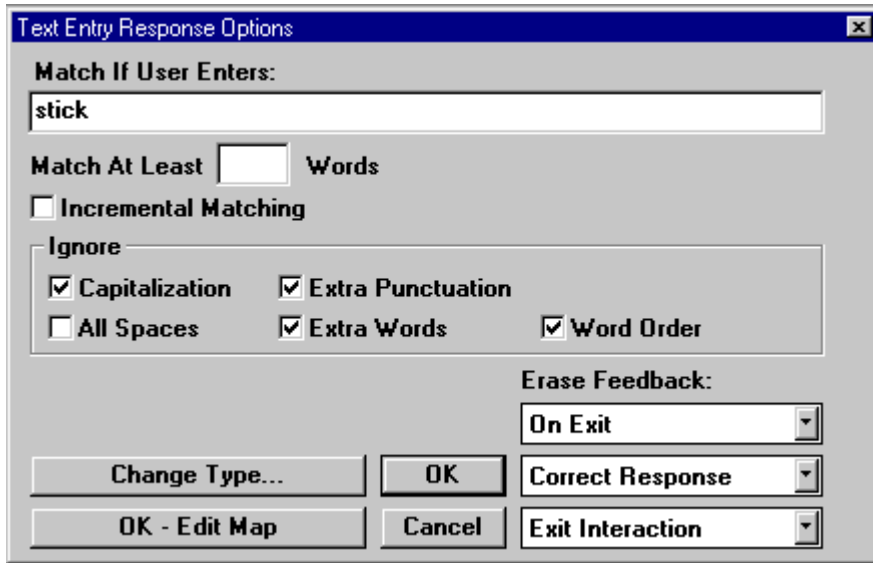
- You should now be looking at something like this:



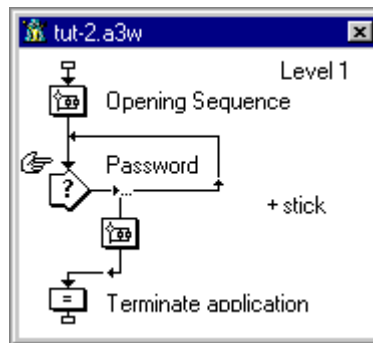
- The “Untitled” label is where the actual password goes, for my examples I am going to use the word stick as the password. Give your group icon a title now - i.e. select your password.
- Now we will let the program know that this is a correct option. To do this double click on the small icon above the group icon you have just placed (circle marks the spot this time):



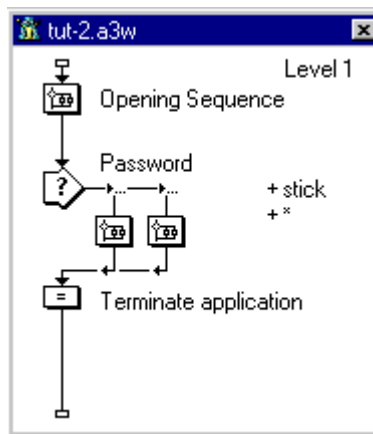
- You will now see the following dialogue box, change the options to match this box (i.e. select “Correct Response” and “Exit Interaction”):



- You will now see something like this:

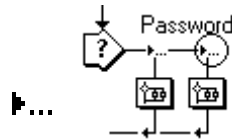


- Notice how since changing the dialogue box entry from “Try Again” to “Exit Interaction” the flow line has changed accordingly. As time passes and you practice and experiment with Authorware you will be able to *read* the flow line, (**top tip:** use good/descriptive icon labels).
- Next we will take care of all the wrong responses.
- First we will take care of the incorrect responses. To do this drag a group icon to the flow line, just to the right of the previous group icon. Name this icon *. The * must come after the password on the flow line, as the * takes everything else. Things should look this now:

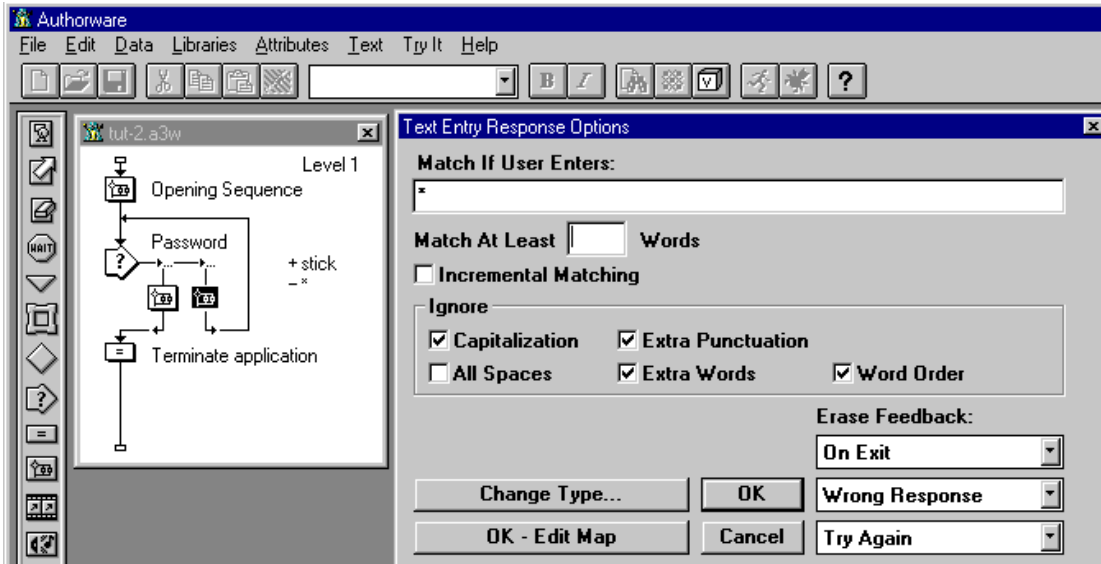


“Make sure yours looks like this!”

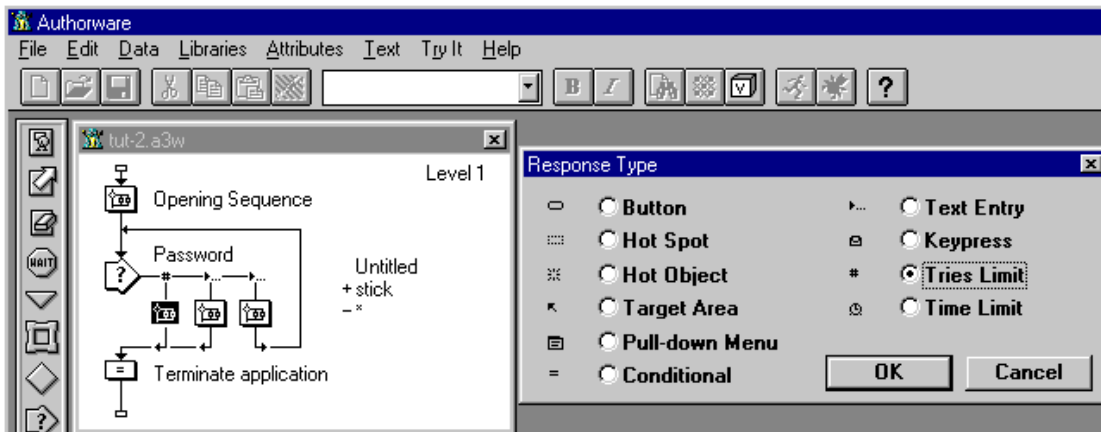
- With this icon we don't want the user to have a single go and exit, we want the end-user to have another go. To do this double-click on the "Response Type" icon:



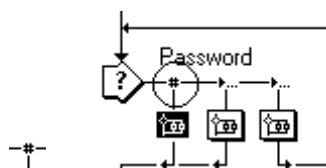
- You should now see a dialogue box like this:



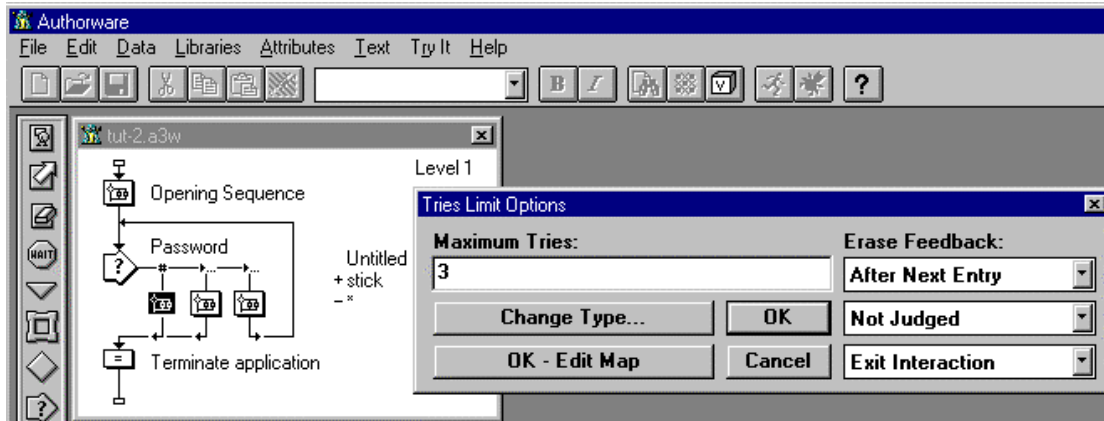
- Alter the dialogue box entries to match this dialogue box. (i.e. "Wrong Response", "Try again").
- Finally we will add the tries limit interaction.
- First, drag another group icon to the flow line, place it before the other two, like this:



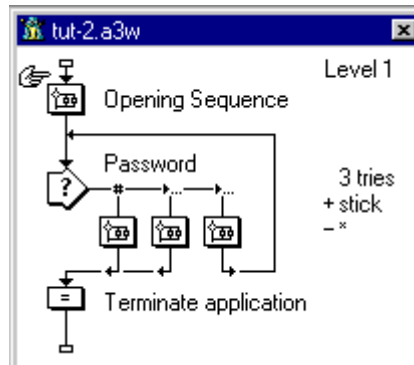
- Select the "Response Type" as "Tries Limit".
- Next, double-click on the "Tries Limit" response type icon, i.e.:



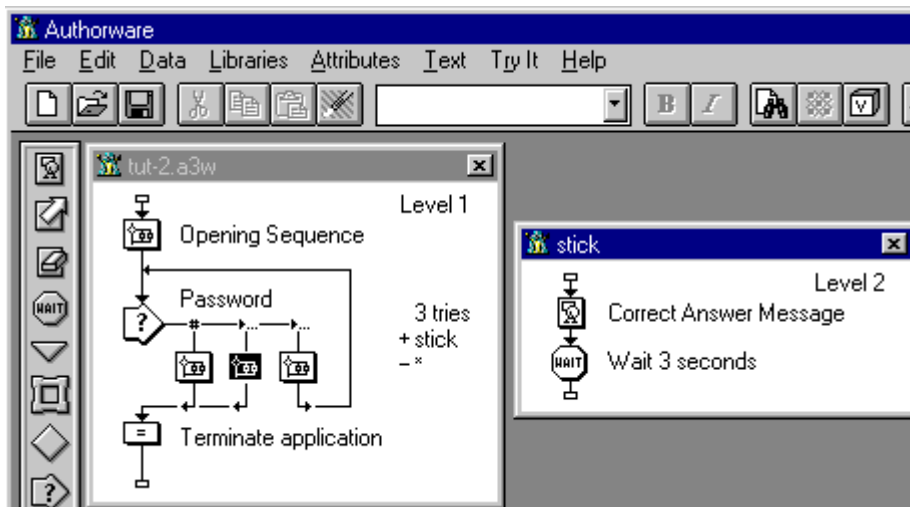
- You should now see the following dialogue box:



- Enter **3** for the number of maximum tries allowed at the password. Don't forget to set up the name for the icon (e.g. "3 tries").
- The **whole thing** should now look something like this:



- You can now run the application. However, you won't see much as you are yet to place more detail within the group icons (i.e. the error messages and the like). We will now add appropriate messages for each of the group icons.
- First we will deal with the correct response. Double click on, and open the password group icon (for me this is the "stick" icon).

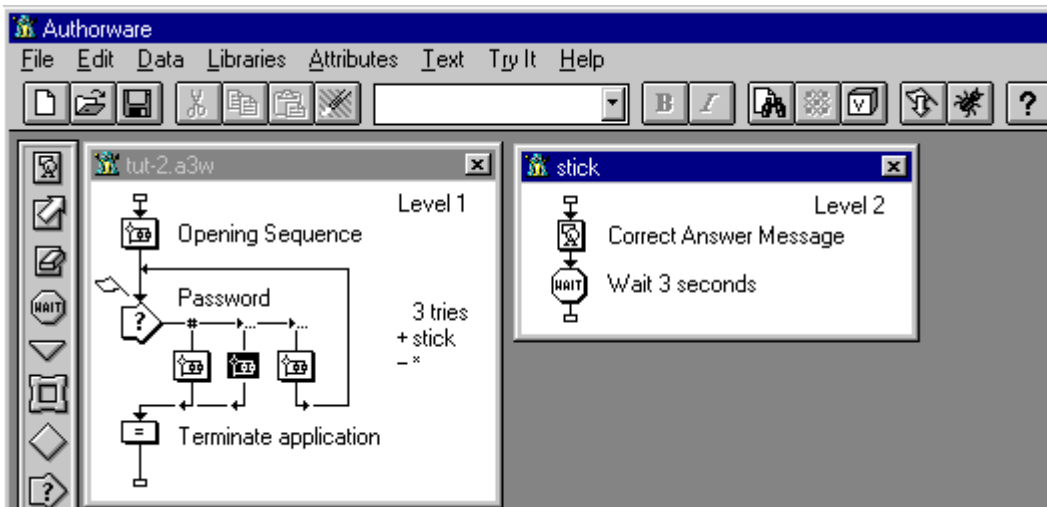


- As you can see I have added a "Correct Answer Message" (i.e. display icon) and a "Wait 3 seconds" (i.e. wait icon). Add these to your own application and *fill* them appropriately.

- You should do the same for all three of the group icons (i.e. add a display icon, with appropriately placed message and a wait icon).
- Run the application and see how the whole thing works.
- You might find it a bit annoying to have to run through the whole application just to get to the bit you want to see running. Authorware has a special function to get around this, the start and stop flags.



- Simply put you place the start flag where you want the program to start running from, and the end flag where you want it to stop (you don't have to use a end flag if you don't want to).
- In the following picture I have placed the start flag at the beginning of the section that we have been working during these last few exercises.



- As yet we have not been using the button bar, but now I will introduce you to run and run from flag icon.



The run icon and the run from start flag icon.

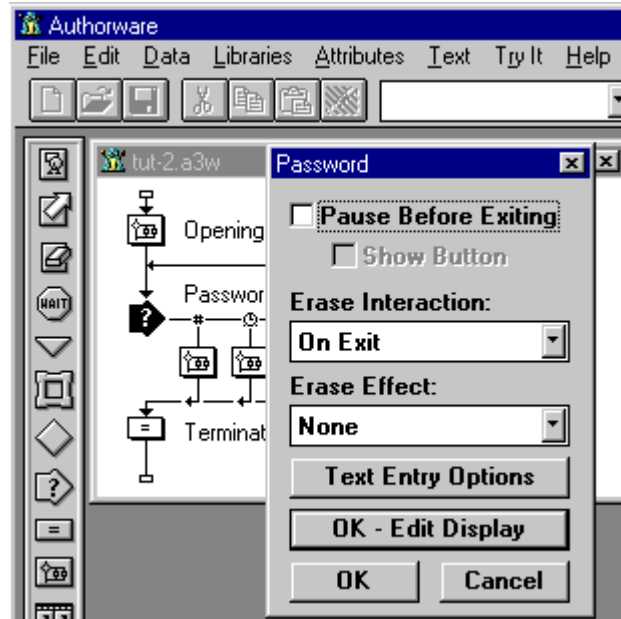


- Obviously, the run from flag icon won't be shown unless there is a flag being used.

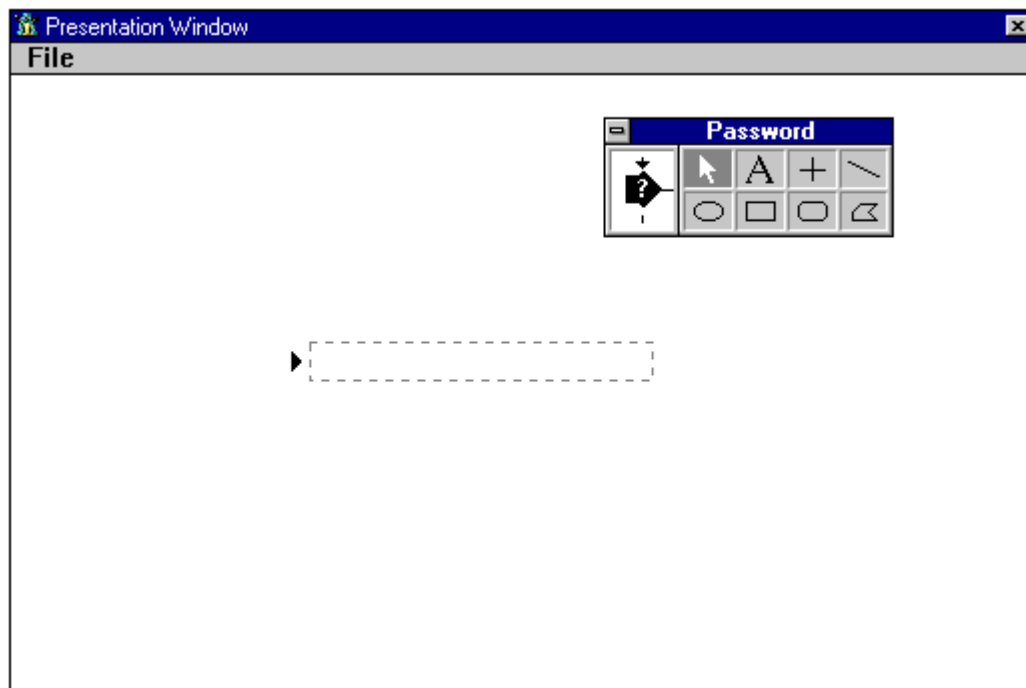
4) More elegant interactions

The program as it stands works but it isn't very elegant. Indeed, it is extremely rough, therefore we will now add some *smoothing* features. We will start with the main interaction icon and then work down the flow line!

- At the moment when the box comes up asking the end-user to enter their password it is assumed that the end-user actually knows that it is a password to be entered. We will now prefix the box with "Password:".
- First double-click on the interaction icon
- Having double clicked on the interaction icon, you will be greeted with the following dialogue box:



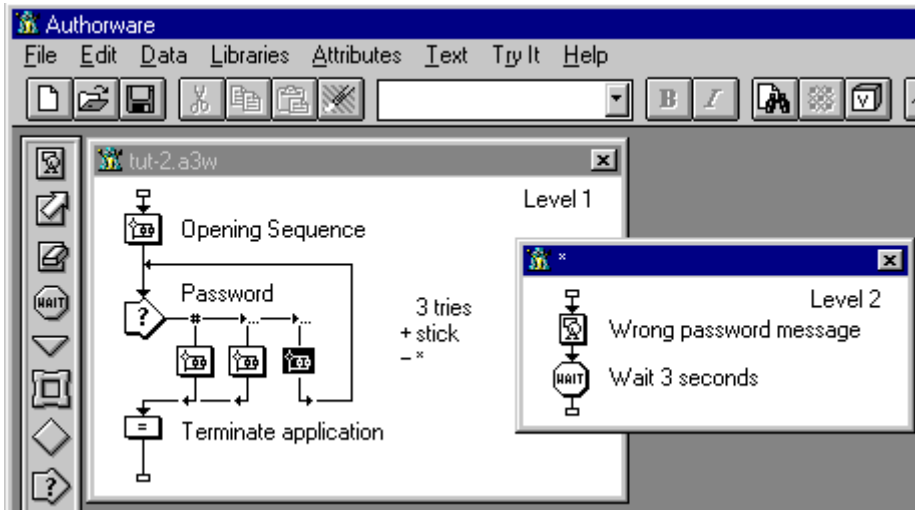
- Click on the button "OK - Edit Display"
- You will now see the customary *display icon* type dialogue box



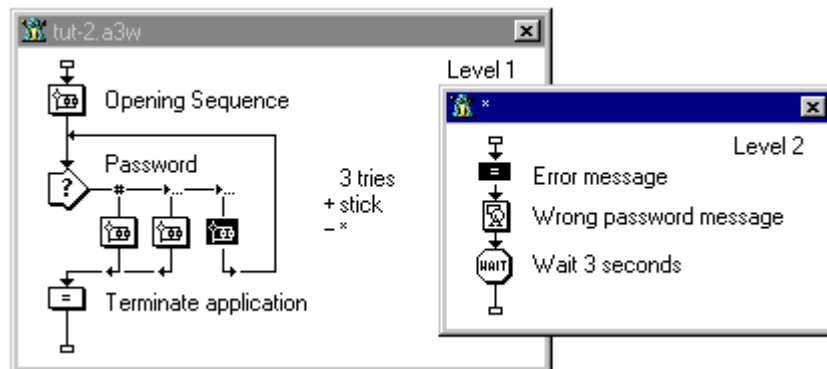
- Edit this display so that “Password:” appears in front of the text entry box, e.g.

Password:

- While you are here you can put a box around the whole password thing to make it stand out if you want, (practice your Authorware drawing skills).
- Next we will look at the wrong password message (i.e. the “*” group icon).
- Double-click on “*” group icon. We will now tailor the error message that you should have already set up in here. You should have something like this on your screen at the moment:



- To improve the error message we will be doing two things: (1) adding a count and using variables;
- (2) tailoring the error message to the number of attempts.
- There are a number of ways to achieve the desired effect, this tutorial will only show one.
- First add a calculation icon above the “Wrong password message” display icon, call it “Error Message”



- Now double-click on the new calculation icon (i.e. “error message”). Enter in the following exactly as it is. You can of course, change the error messages and anything else once you understand what is happening.

```

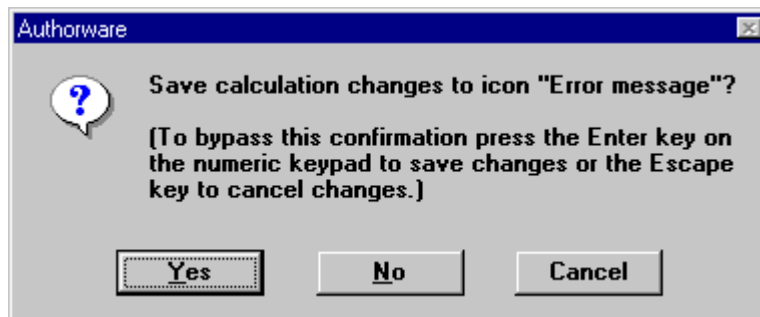
Error message
wrong_password_count:=wrong_password_count+1

if wrong_password_count = 1 then
  error_message := "1st wrong password - please try again"
else if wrong_password_count = 2 then
  error_message := "2nd wrong password - LAST ATTEMPT - get it right - please try again, "
end if

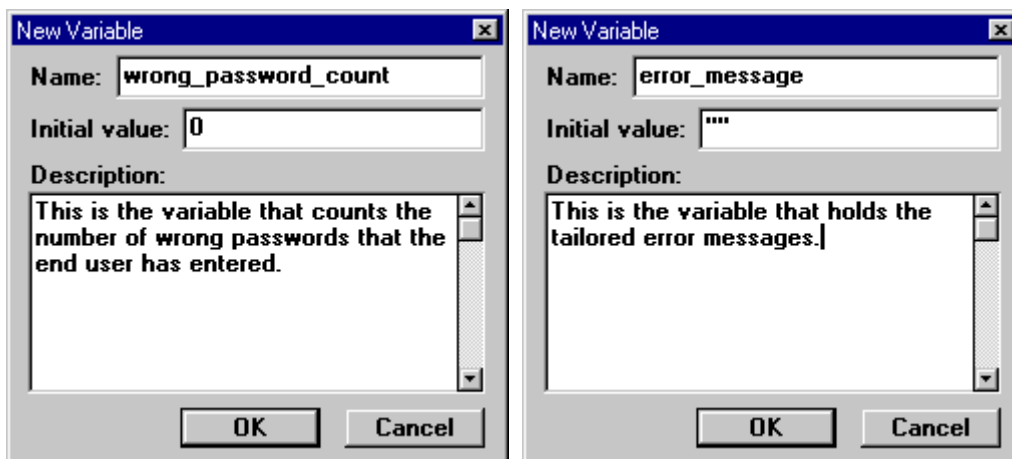
```

“Be careful when copying the details from the above box!”

- You should be able to read through the above and work out what is happening. Basically a count of the number of wrong passwords entered is being set up. As the number of wrong passwords goes up the error message is altered appropriately.
- Once you have entered the above section of code close the calculation icon dialogue box (CTRL+F4).
- When you select close, Authorware will then look at the ‘code’ that you have just entered and check that it understands what you are asking it to do.

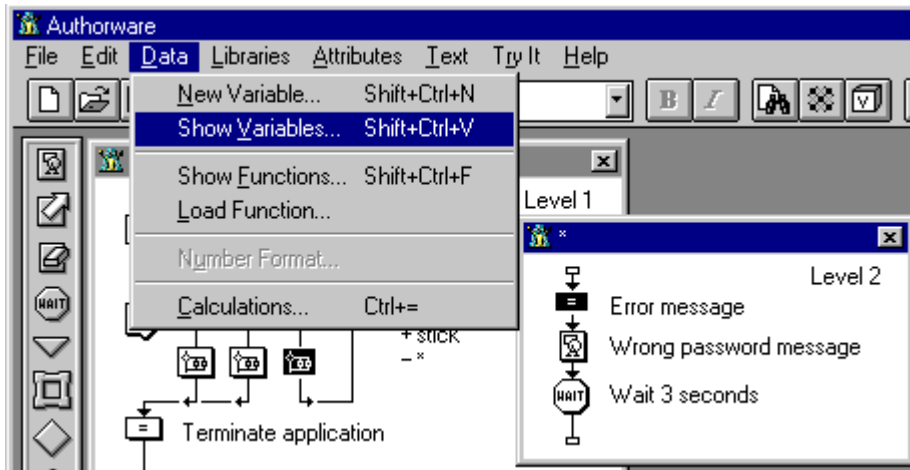


- Select “Yes” - Authorware will now check the code. As Authorware works through the ‘code’ it will look at the different variable to check that it know them. If it does not it will as you to enter in details about them:
- Copy the details in the following dialogue boxes:

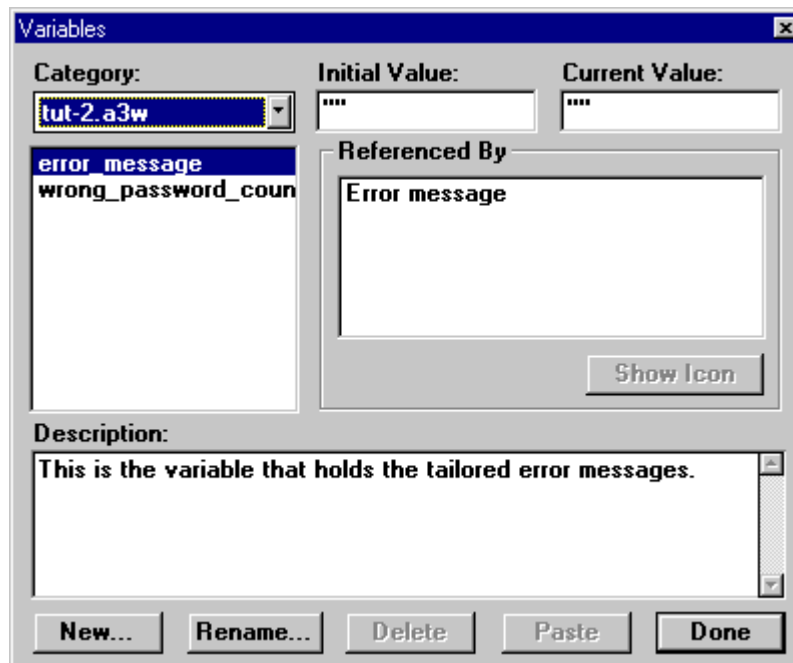


- As with the function “Quit” that we looked up (calculation icon page 14) you can also look up variables.

- Go to the menu, “Data” then “Show Variables”, or press SHIFT+CTRL+V

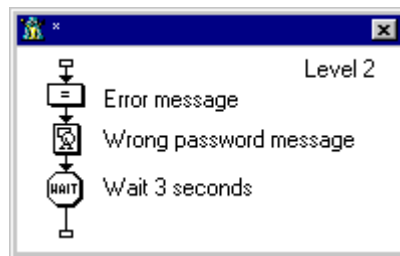


- You will now see the following dialogue box:



- Your category might be different. Select category, and choose the option which is the same name as you application. In my case it is “tut-2.a3w”.
- You will then see a list of you two new variables.
- It shows the Initial value that the variable is set to and the current value, (this changes as the program is used). You can edit any of these values.
- Also of interest is the window called “Referenced By”. This windows shows all the icons that make use of the variable, this is very useful for *debugging* programs. If you click on “Error message” and then the “Show Icon” button it will show you the icon in question. You may find this feature useful when developing your own applications.
- Now we will make use of the “error_message” variable in the “Wrong password message” display icon.

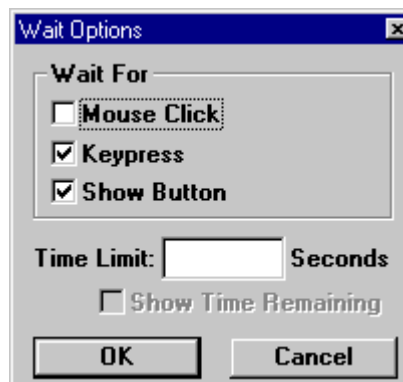
- Double-click on the “wrong password message” display icon



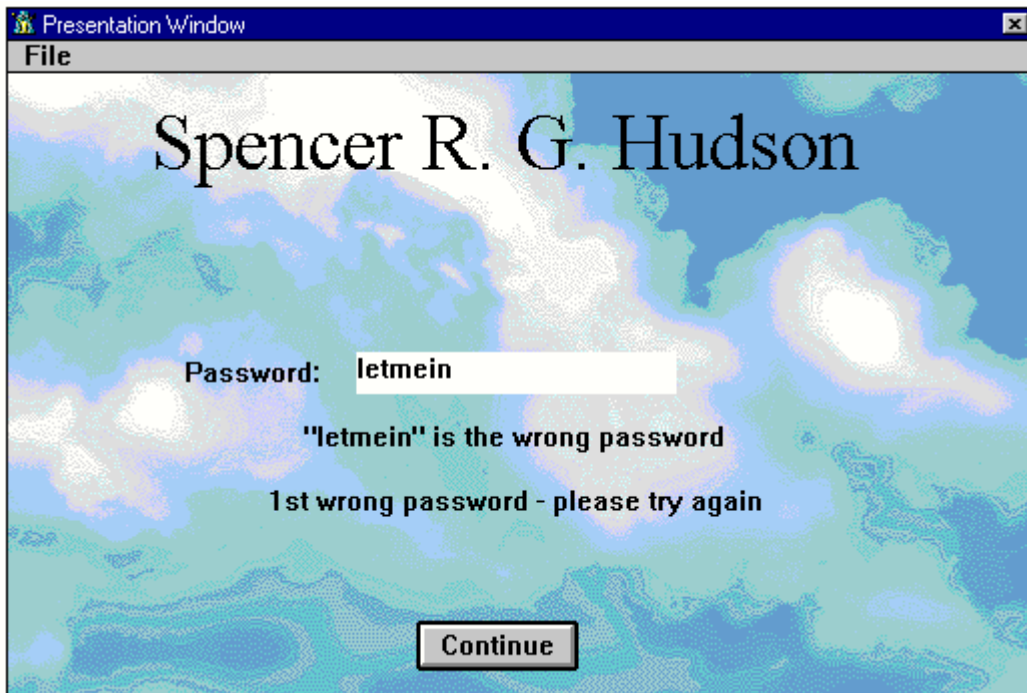
- We will now change the current message to be the “error_message”. Displaying a variable in Authorware is very simple, you just place the name of the variable in a text element surrounded by curly brackets (i.e. “{error_message}”).
- Things should now look like this:



- Before we go on, we should check the position of the error message and the button for the pause.
- Run the application and enter an incorrect password.
- Press CTRL+P when you get the error message up!
- I have assumed that you have configured the wait icon as follows:



- You should now have something like this on your screen:



“Be careful to select an appropriate background!”

- Press CTRL+P to pause the application, and now move the elements around on the screen. I have moved the “Continue” button to be at the bottom middle!
- Already the program is a lot more end-user friendly - just by simply creating more useful error messages.
- Having see the error message it should be erased. So now add an erase icon after the wait icon that removes the error message.




- Don't forget to give the erase icon an appropriate label.
- Run the application NOW.
- If you see room for any visual improvements make them as you go along, (i.e. CTRL+P and: move the elements around the screen; give them a different effect, colour...)

In your own time:

In general

- Examine the “File Setup” option, how many icons have you used (icons used so far: _____), you have a maximum of 50 with Authorware Working Model.
- Notice how big the file is (current size: _____). Now save the file and compress it and go back to the “File Setup” menu and see how big the file is now (new compressed size: _____). Do you notice any difference? This is one of Authorware’s more annoying features - you have to opt to compress the file, it is not done by default when you save it, so keep an eye on you file size.



- Colour code you icons with:  It’s located at the bottom of the icon/tool bar. To use these colours just select the icons that you want to give a colour then click on a colour! Couldn’t be easier!

The Sound Icon

- Experiment with the “Wait Until Done”, “Concurrent”, and “Perpetual” settings. Note the perpetual setting will only work if you have “Play Sound” options set. For example if you select “perpetual” and “Until True” the sound will continually play!
- Look under the “Data”, “Load Functions” for instructions on how to play a MIDI file. This is not a simple operation and you will require the use of a calculation icon! Ask for more details.

Calculation Icon and variables

- Experiment with the error messages and the number of tries. See what different combinations of messages can be made.
- Don’t limit your self to using variables in the error message - where else can you use them?
- Rather than using the calculation icon itself you can attach a calculation to an icon. This way you do not use a actual icon. To do this simply select the icon you wish to have a calculation before and press CTRL+=. You will then see the normal calculation dialogue box. Enter in you calculation and close as normal.



verses



- You can now add another enhancement to the interaction by clearing the password entered by the end-user. That is, add another calculation icon (after the erase icon which says EntryText:=” This will have the effect of clearing the old password from the screen.
- Top tip 3: Good user interfaces are all about the little touches!

Interaction Icon

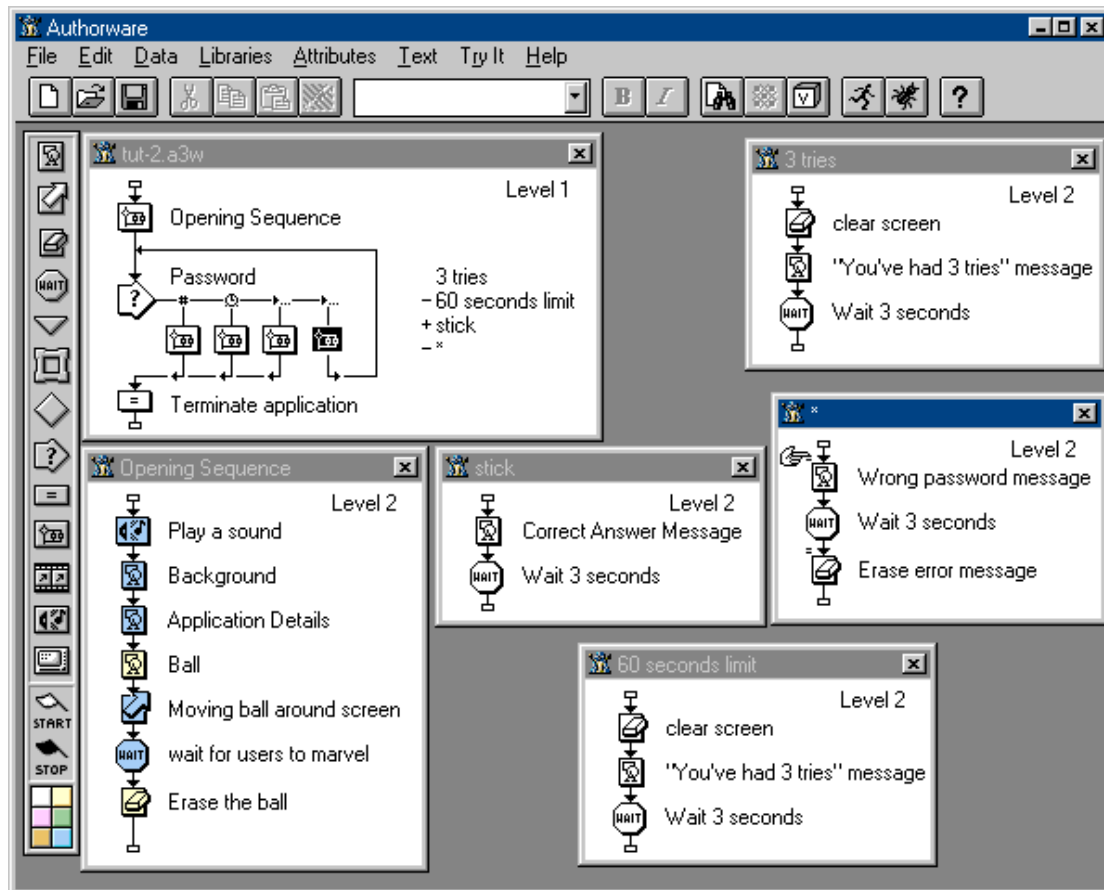
- Text Entry: You can alter the format (e.g. colour, size, font) of the text being entered in the normal ways. To do this double-click on the interaction icon, select “OK - Edit Display” and then select the text entry box. Now apply the formatting options that you want.
- Text Entry: In the text entry dialogue box there is more than just the simple one word matching facility - you can look for case sensitive, single word in a sentence, order of words... Experiment with these extra features and with your chosen password (try making it two or three words/numbers)

- Time Limit Option: Similar to the “Tries Limit” option, time limit as its name suggests is a limit on the time available to the end user to perform an interaction. Add this to the *password* interaction, set the limit to 60 seconds.

Summary

- Use flags (start and stop) to control which parts of the program you wish to run.
- Pause and edit screen: CTRL+P
- Calculation on an Icon: CTRL+=
- Using variables: {variable_name} use underscore only
- Group icons together with: CTRL+G
- Top tip 1: use descriptive icon
- Top tip 2: colour icons for identification purposes
- Top tip 3: good interfaces are all about the little touches

The whole application (with the time limit) should look something like this now:



A Deliberate mistake has been in this application can you spot it. Run the program a few times and see if you can see where it is. I will give you a clue it is to go with the interactions.

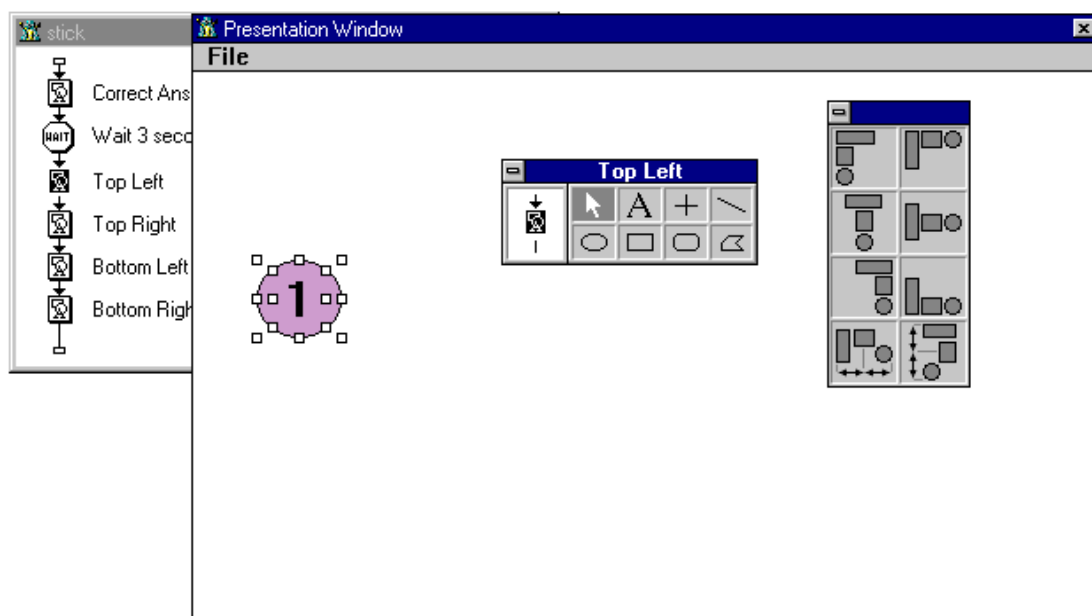
3rd Tutorial - Building a simple game/puzzle

In this tutorial we will build upon the work already done on the interaction icon. This time we will be focusing upon the “hot spot” interaction type. We will be creating a simple drag and drop puzzle. In essence there will be four shapes and the end-user will have to drag them to the correct position.

- We will first have to create the pieces of our puzzle. Nothing special is required but the more interesting the better. For simplicity I will just be using circles with number in them.
- We should be starting with something like this:

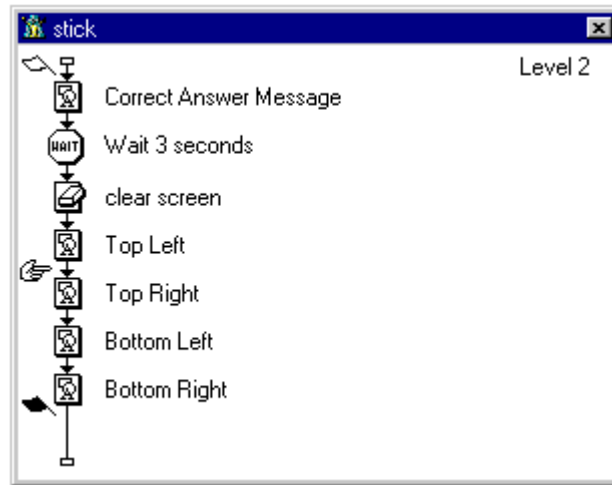


- Add an erase icon after the wait icon to clear the screen! If you have difficulty deciding what needs to be erased don't select anything, just run the application. When the program arrives at this empty icon it will pause and allow you to fill it in. Then all you have to do is select all the elements you want to be erased!
- Now we will add out four pieces of puzzle, which are to be moved into the correct position.
- Drag and drop four display icons onto the flow line, one after another. All after the wait icon. Label them Top Left, Top Right, Bottom Left and Bottom Right. The names correspond to their places on the puzzle.
- In each of the display icons add the appropriate image for the puzzle, e.g.:

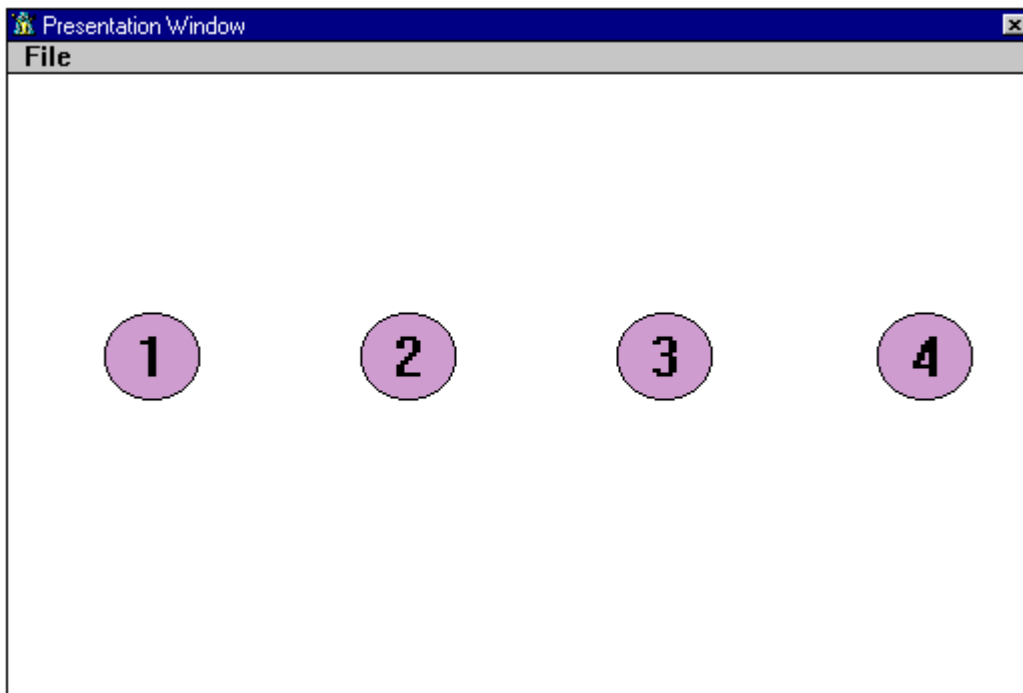


- If you are copying my example, don't forget to align the number within the circle (align dialogue box: CTRL+H).
- Create the other three pieces of the puzzle in a similar way and arrange them horizontally across the page.

- So you should now have something that looks like this:



- When I run the applications (notice I have set my flags), I get the following screen. Note, it is built up from the four separate display icons showing they different pieces of the puzzle.

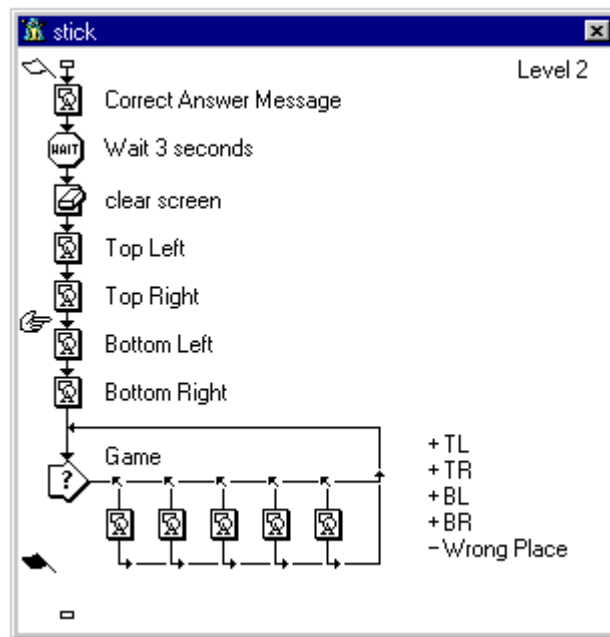


- Now that all the pieces of the puzzle are ready to start building the actual puzzle part of the application.
- Drag a interaction icon just below the last puzzle piece, label it “game”.
- Next drag a display icon to the flow line, and select “target area” as the interaction type.

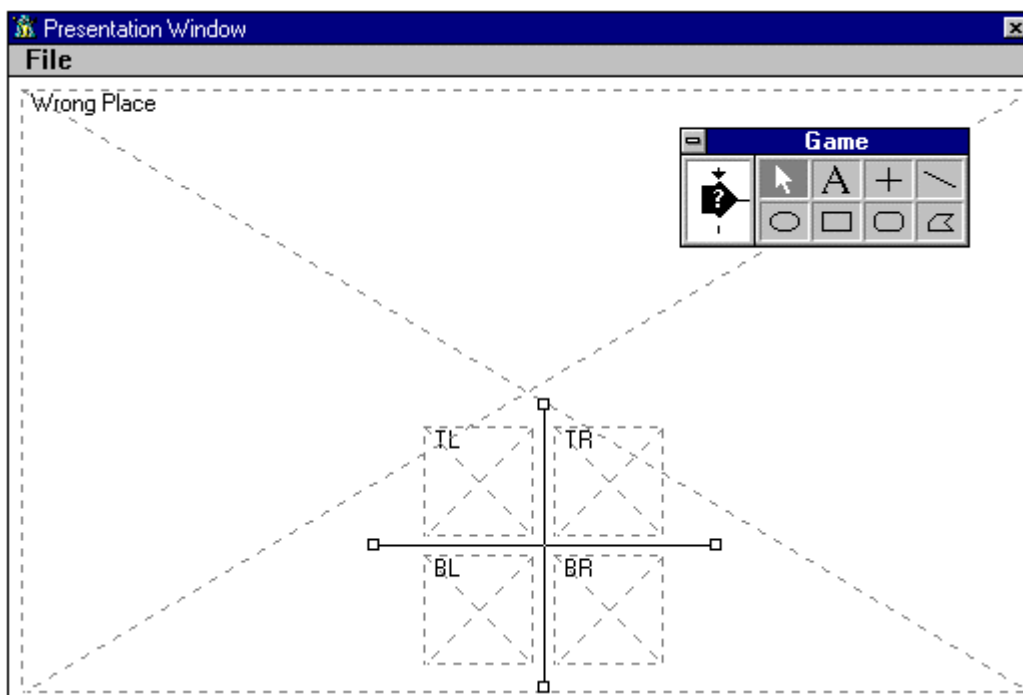


- Label the display icon TL, for top left.
- Do this three more times, once for each of the other puzzle pieces.
- Label each of the display icons TR, BL BR respectively.
- Finally add one more display icon, and label it wrong place (wrong response).

- You should now have something like this:

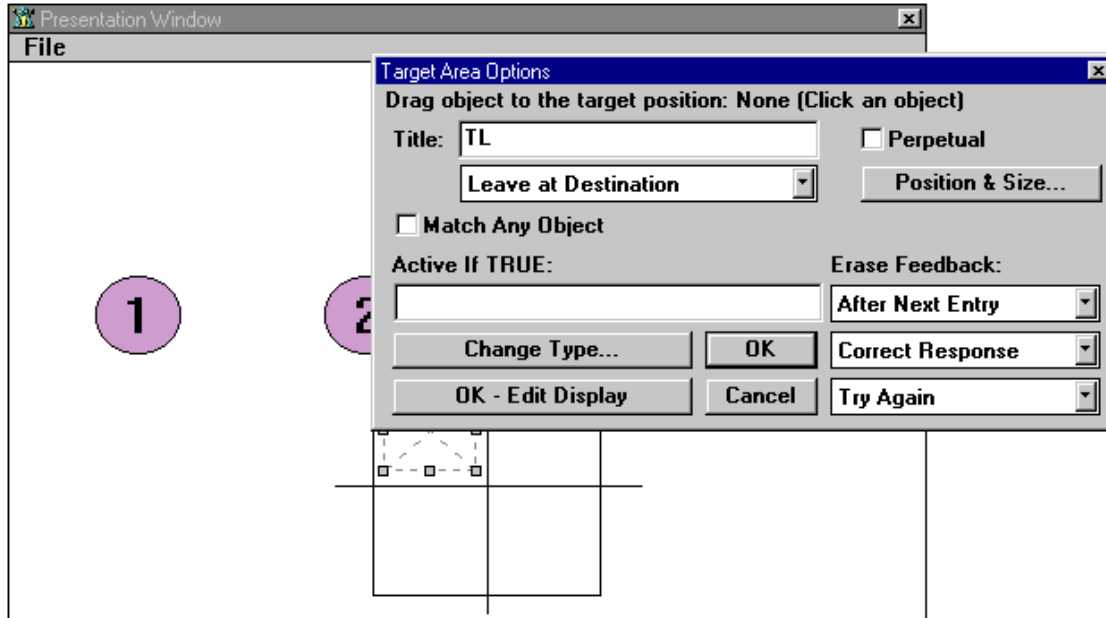


- To help people with the puzzle we will set up the board where the pieces will go.
- Double click interaction icon and select "OK - Edit display"
- Set the display to look like this (note: I have added the two lines which are crossed, thus creating four areas within which the object can be placed, this is to aid end-user position their puzzle pieces):
- Top tip: to move a target click on the boarder (i.e. dotted line), to size a target area click on the lilt boxed on the boarder lines.

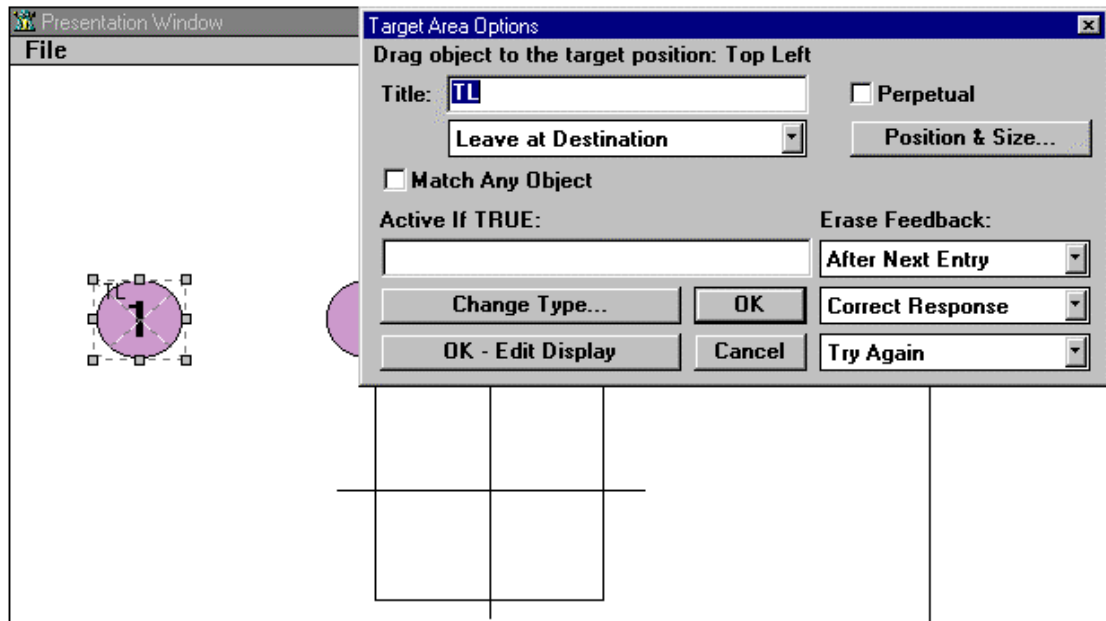


- We will now go through each of the icons and configuring target areas to respective objects.
- Providing the flags are in the correct place, run the application!

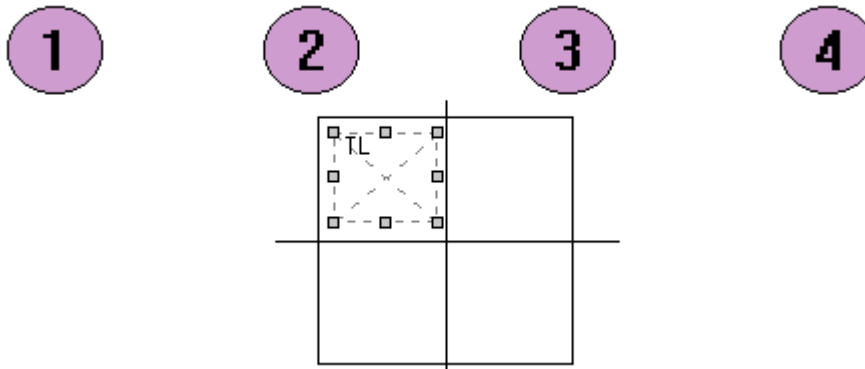
- You will eventually see this screen. On this screen Authorware is asking use to identify the object that is associated with the “TL” target area.
- To associate them together simply click on the top left object.



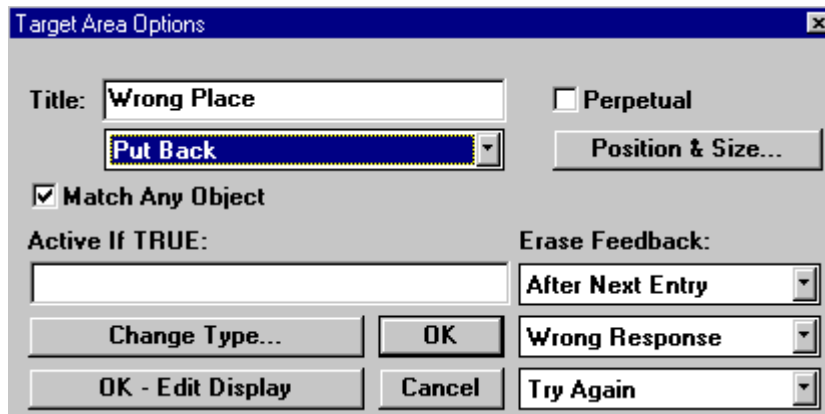
- As shown in the next picture, when you do associate the two together the “Target Area” moves.
- Move it back to the top left!



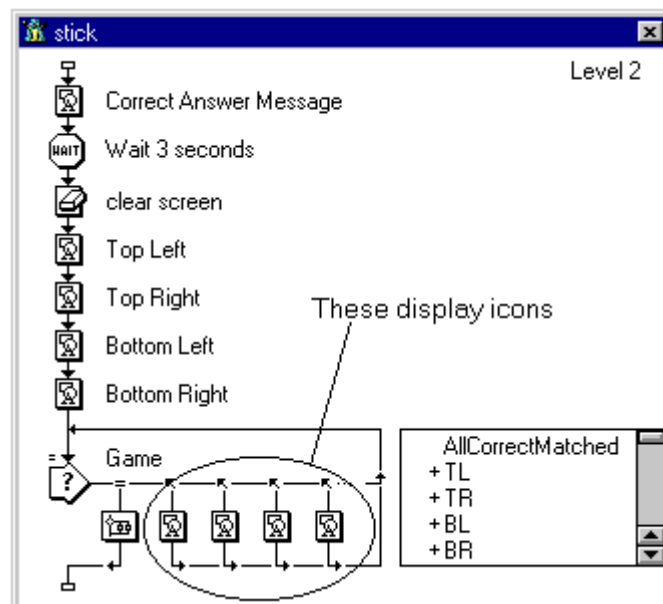
- Having moved the target area back things should look like this:



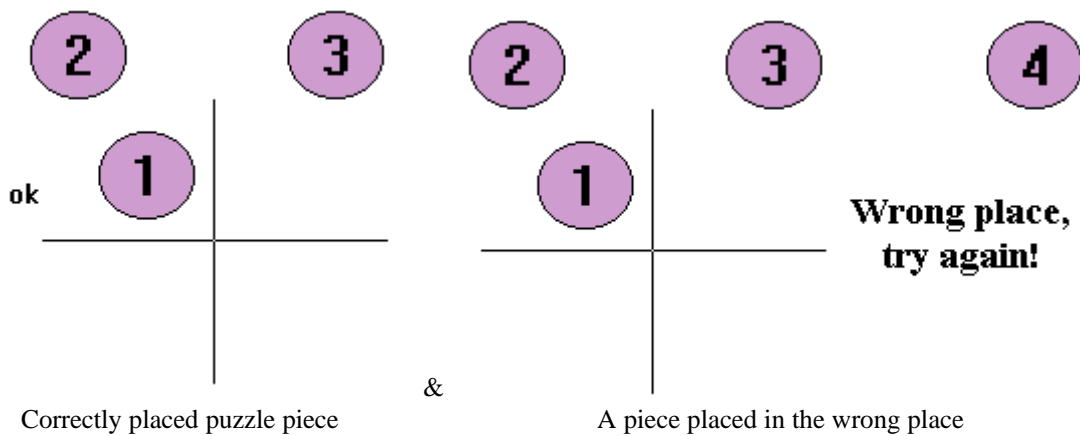
- Repeat the process for pieces 2, 3, and 4.
- For the “Wrong place” target area we do something a bit different.
- The following dialogue box contains all the details. Things to note are: (1) “Put Back” that is if an object is placed in the wrong place it is *put back* to its original place; (2) “Wrong Response”.



- Next we will put *appropriate* messages in each of the display icons, e.g.:
 - TL ~ Top left: position the word “OK” just next to the top left box
 - TR ~ Top right: position the word “OK” just next to the top right box
 - BL ~ Bottom left: position the word “OK” just next to the bottom left box
 - BR~ Bottom right : position the word “OK” just next to the bottom right box
 - Wrong place: position the words “wrong place” somewhere on the screen



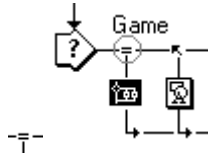
- The following two pictures shown what the feedback message (when the application is running) should look like. These are the messages that you have *just* placed in the display icons:



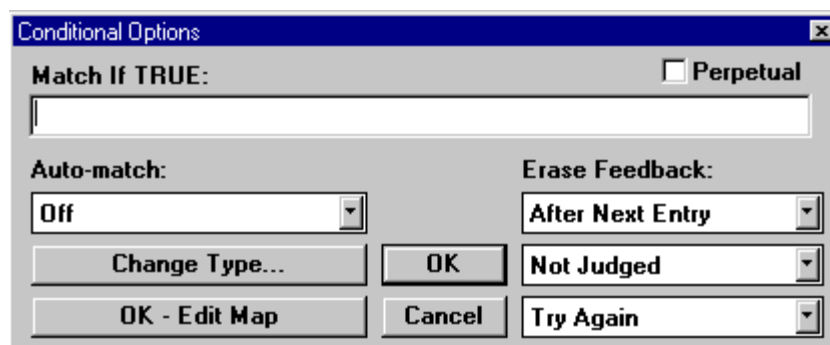
- Run the application now, you will notice that there is no exit. That is, even after placing all the pieces of the puzzle in the correct place nothing happens.
- We will not add a final icon to the puzzle that will permit exiting of this part of the application.
- Drag and drop a group icon to the interaction icon, place it before all the other icons.
- Give it a “Conditional” response type:



- Double-click on the response type icon (circle marks the spot):

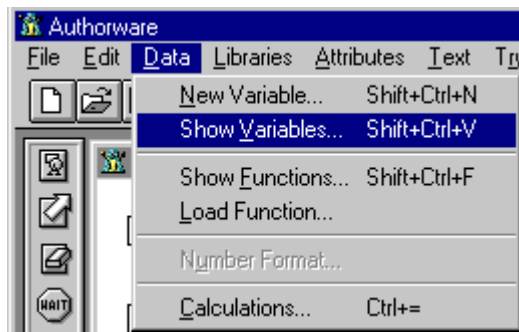


- You will now see the following dialogue box:

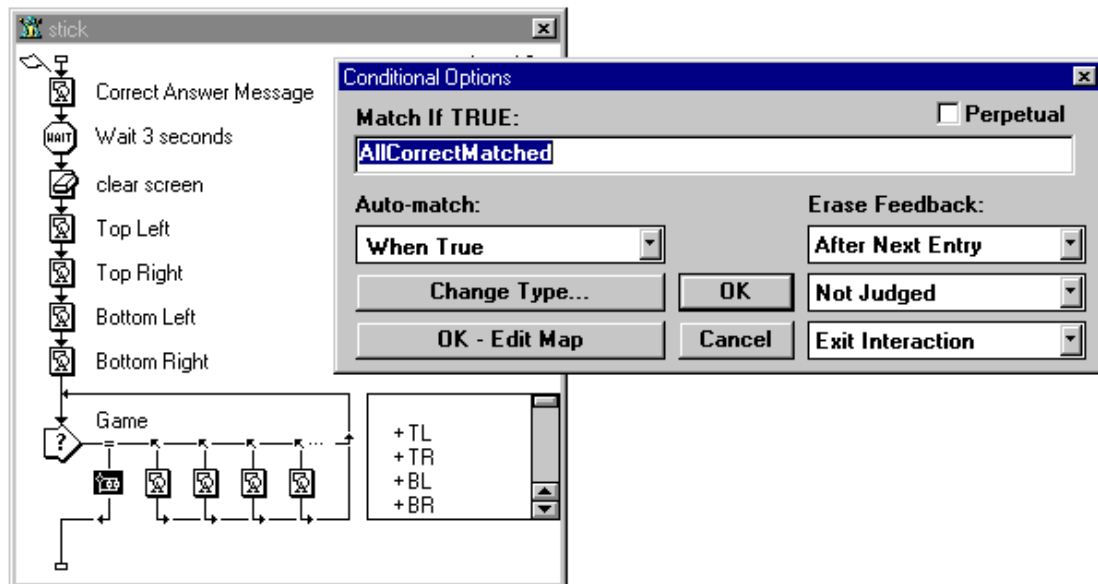


- We want to “Match if TRUE” that all the pieces of the puzzle have been put in place. Authorware provide a special variable for the.

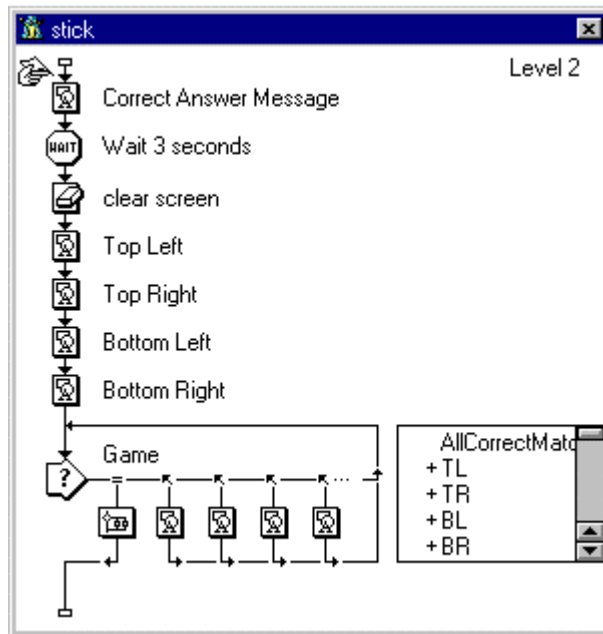
- As you may remember we can search for variables:



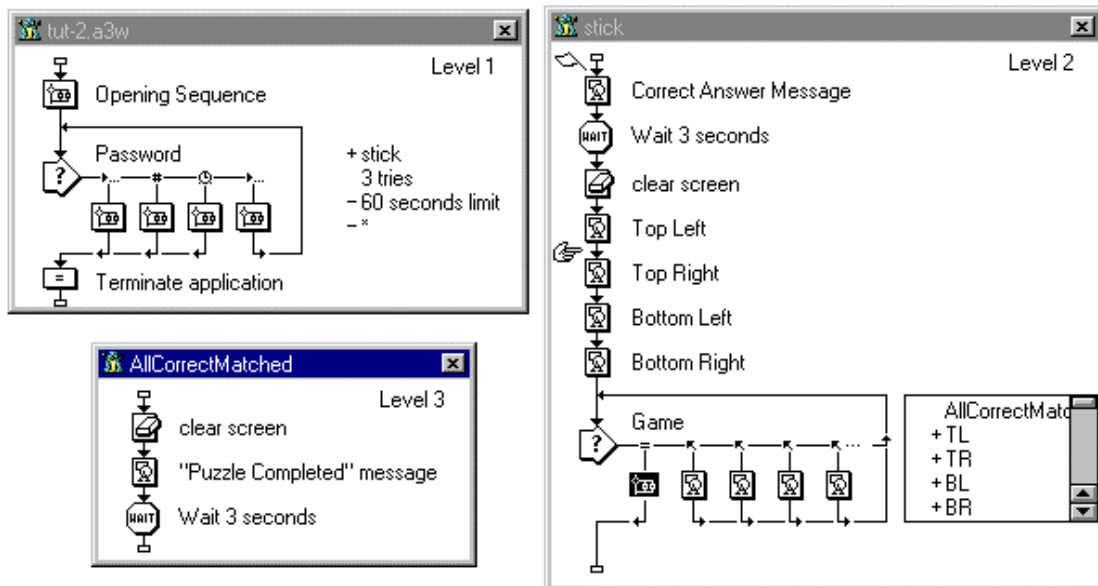
- Open the “Show Variable” dialogue box and search for **AllCorrectMatched**. Paste this into the “Match IF TRUE”. AllCorrectMatched is a special Authorware variable that checks that all the “correct” elements are true.
- Copy the details from the “Conditional Options” dialogue box below:



- You should now have a flow line that looks like:



- One last thing is now left to do, and that is to give an appropriate feedback message when the puzzle is completed.
- Double-click on the “AllCorrectMatched” group icon and add the usual icons (i.e. erase, display, wait, erase icons).
- The application should now look something like this:



Notes

- The deliberate error from the first set of exercises has been corrected in this picture. The error was to do with the number of attempts that could be made at the password. The error was that only two attempts could be made, this was because the “Tries Limit” option was positioned before the “Text Entry” response option.
- Always be very careful when positioning your interaction types.

In your own time

In General

- Examine the “File Setup” option, how many icons have you used (icons used so far: _____), you have a maximum of 50 with Authorware Working Model.
- Notice how big the file is (current size: _____). Now save the file and compress it and go back to the “File Setup” menu and see how big the file is now (new compressed size: _____). Do you notice any difference? This is one of Authorware’s more annoying features - you have to opt to compress the file, it is not done by default when you save it, so keep an eye on you file size.

Calculations

- Add a calculation that keeps track of the number of times it takes the end-user to place all the pieces of the puzzle, (this needs to be done for the next tutorial).

Interaction Icons

- Target Area: Experiment with the “Put back”, “Leave at destination”, and “Snap to centre” options. See what effects you can get. Also, experiment with the “Position & Size...” button, you can define precisely how large and where a target area is. By using both of these features it is possible to create very accurate target area interactions.
- Tries Limit and or Time Limit: With the puzzle we can add a tries limit and or a time limit to the interaction - experiment with these!

Summary

- Align elements: CTRL+H
- Top tip: to move a target click on the boarder (i.e. dotted line), to size a target area click on the lilt boxed on the boarder lines.

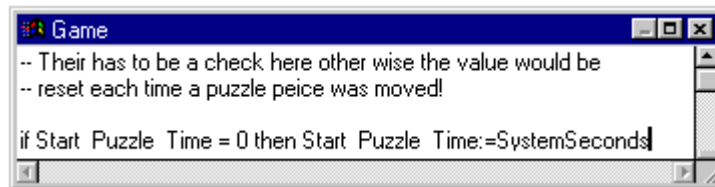
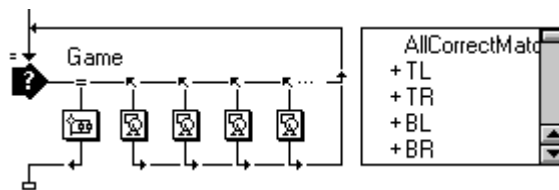
4th Tutorial - Scores and Hypertext Help System

This tutorial will investigate two different features of Authorware. The first is a scoring mechanism, the second is a hypertext help system. The first of these is building upon work done in a previous exercise (i.e. a simple counter variable was used to count the number of tries it took the end-user to complete the puzzle). The second mechanism is a new Authorware command and possibly one of its most useful.

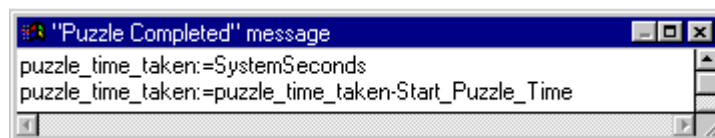
1) More elegant interactions - timer facility

We will begin by adding a timing facility to the puzzle, this will require the addition of a couple of calculation statements (not icons i.e. CTRL+=).

- First we will need to configure and start the timer. Authorware has a special variable to help us with this, it is called "SystemSeconds". It contains the number of seconds since the computer was started or restarted. We will use this as a counter - to do this is very simple, we make a note of SystemSeconds when we start the activity (e.g. 10) and then compare it to the value when we stop the activity (e.g. 25). Take the two away from each other and we get the elapsed time (e.g. 15 sec).
- To begin we will add a calculation to the Game interaction, this is where we will make a note of the "SystemSeconds":



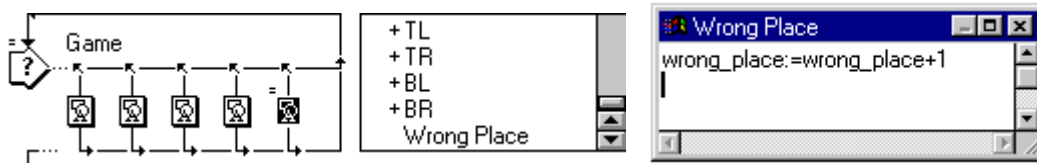
- Next we will add its counter part which calculated the actual time spent doing the puzzle.
- Add another calculation this time to the "Puzzle Completed" display icon:



- For this all to work the final step is to alter the "Puzzle Completed" display icon so that it actually displays the time taken:

Well done you have completed the puzzle!
 Number of incorrect tries: {wrong_place}
 Time taken: {puzzle_time_taken}

- You should have already setup the {wrong_place} counter. If you have not then add a calculation (i.e. CTRL+=) to the **Wrong Place** icon. Enter the following calculation:



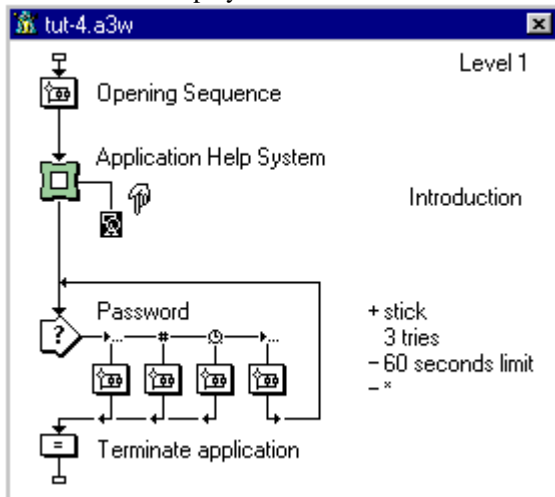
2) A Hypertext Help System

We will now add a simple hypertext help system to the application. This a contrived example and as such won't really add much to the application. It is purely designed to show you an example of the **framework** icon, and the hypertext command. Both of these *tools* are very useful in creating functional applications



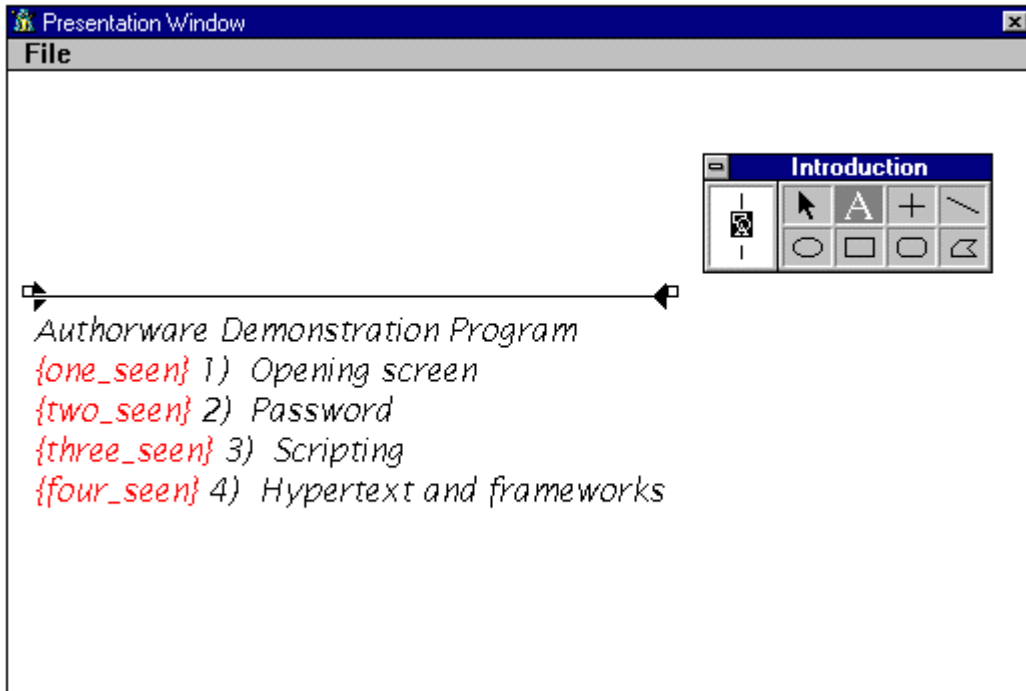
The framework icon

- First of all add a framework icon to the flow line, add it in the position as shown below.
- Also add a display icon.

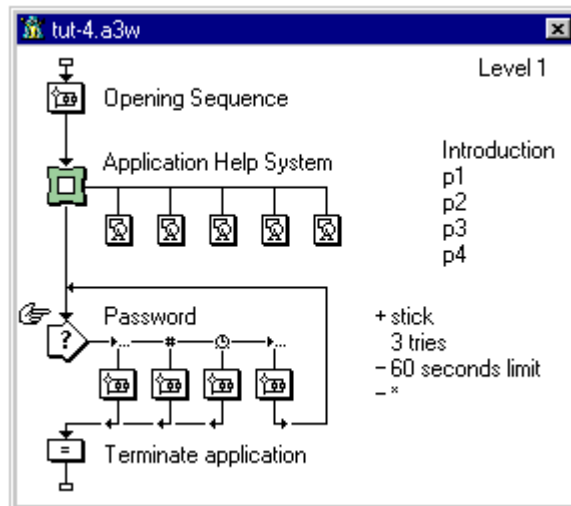


- Shown below are the details that I have entered into the display icon.

- Four variables have been added to the text display (i.e. {one_seen}, {two_seen}, {three_seen} and {four_seen}), these are flags to indicate if a end-user has seen that page, (this type of feature is considered to be *end-user-friendly*).



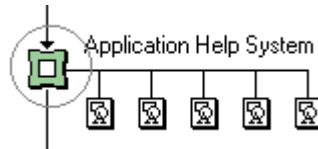
- Now drag another four display icons to the framework icon.
- You should now see a flow line like this:



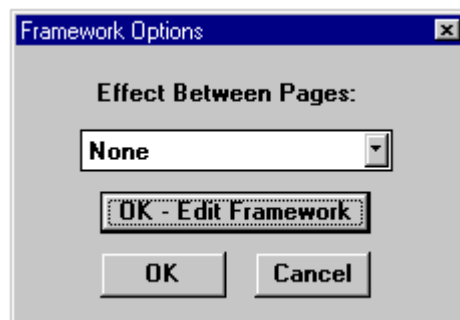
- Note: If you have been checking the number of icons that you have used you will now that we have now reached our limit of 50 icons. Therefor if we need to add anymore we will have to cut back somewhere?
- Now, add some simple text into the four display icons. Along the lines of the four subject topics: (1) opening screen; (2) password; (3) scripting; and, (4) hypertext and frameworks.

There are now two things left for us to do, first we will now add the framework controls, and then we will add the hypertext facilities.

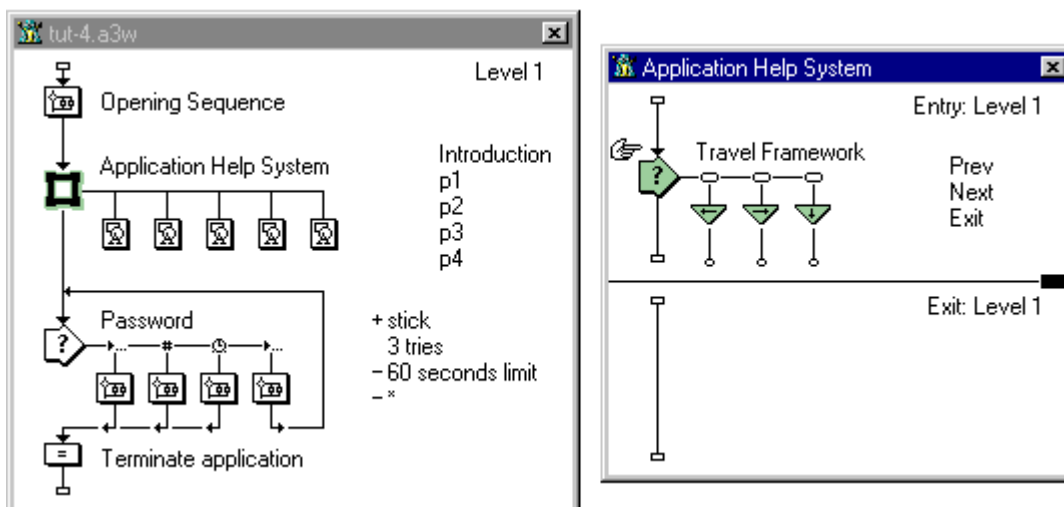
- First, let's get the framework controls to work.
- Double-click on the framework icon:



- You should now see the following dialogue box:



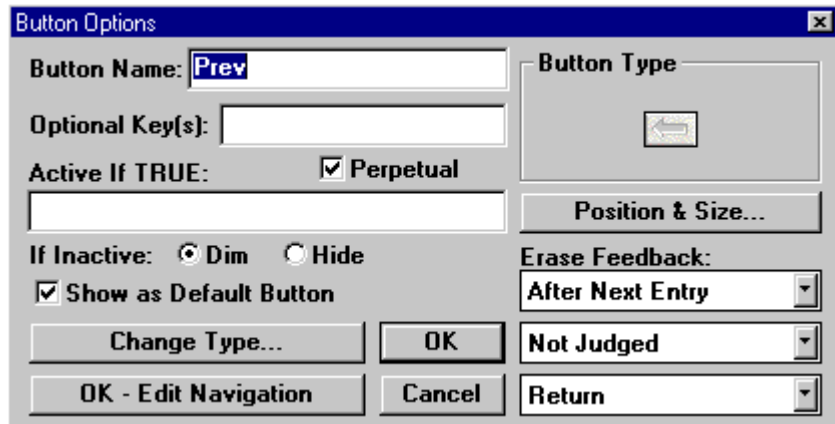
- Select the button "OK - Edit Framework"
- You can if you want select a transitional effect between pages, (e.g. smooth change)
- You should now see the Framework Dialogue box:



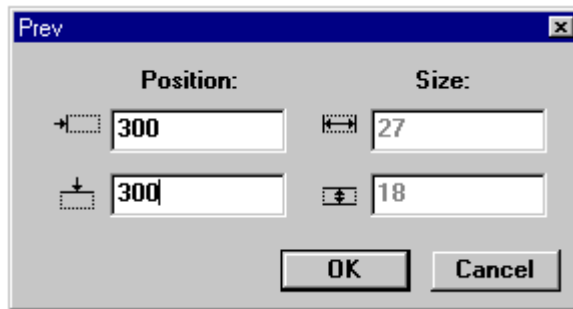
- Basically the framework icon can be considered to be a group of navigation icons (as shown in the diagram above on the right hand side - i.e. contained within the framework icon), which are applied to a number of other icons. In the example what we are doing they are display icons. The framework icon can save a great deal of *programming/effort*.
- In the above dialogue box on the right hand side three buttons are defined: (1) **Prev**; (2) **Next**; and, (3) **Exit**. We will now position these on the screen.
- Double-click on the button icon for the **Prev** button.



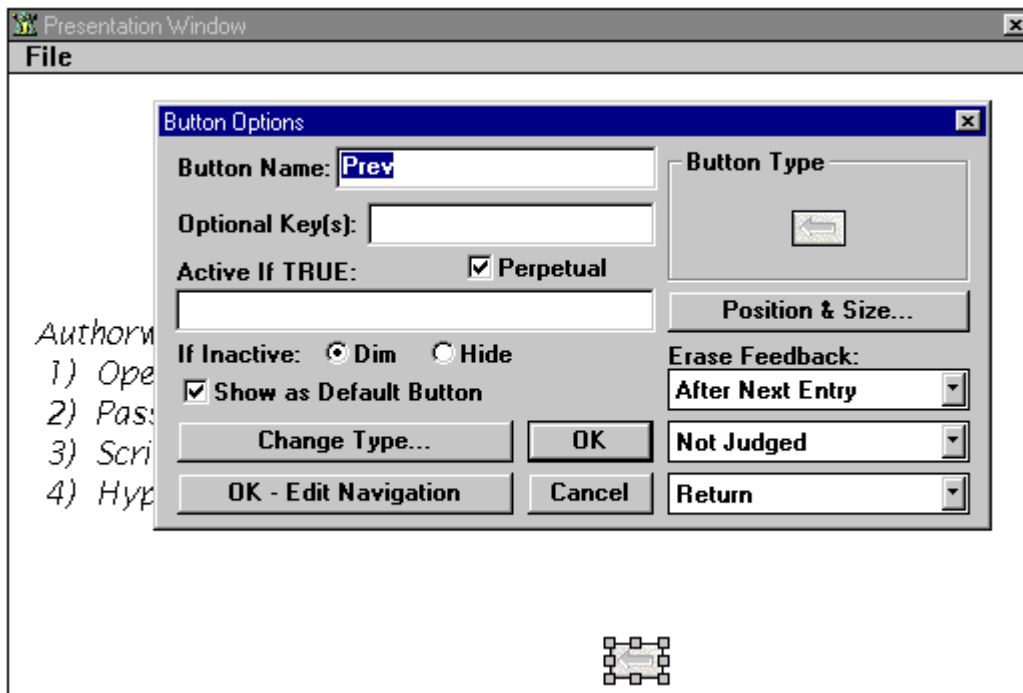
- You will now see:



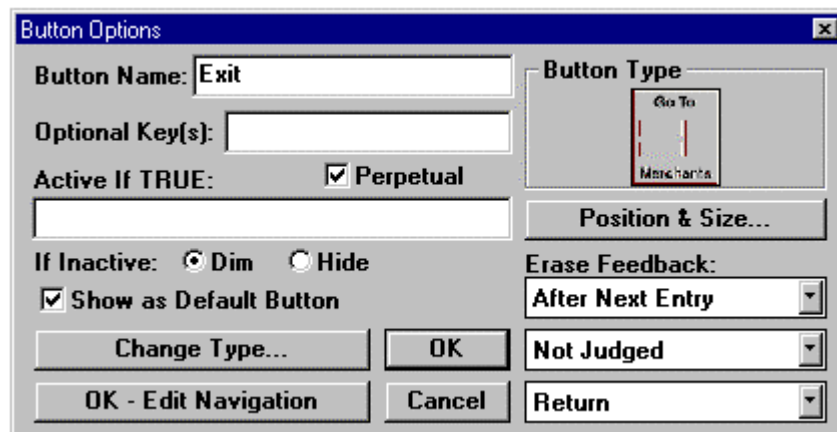
- Notice, the button in the top right of the (**Button Options**) dialogue box. This is the **Prev** button that will be shown on the framework screen that is applied to all the other display icons. At present the icon is drawn off the screen (this is an error with the default positioning), therefore we will have to manually position it.
- Click on the '**Position & Size...**' button, you should now see a new dialogue box with size and position options.
- Alter the dialogue box to look like this (300x300 is within the current display icon screen size):



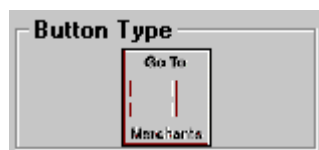
- You should now be able to see your **Prev** button, position it on the screen as shown below.



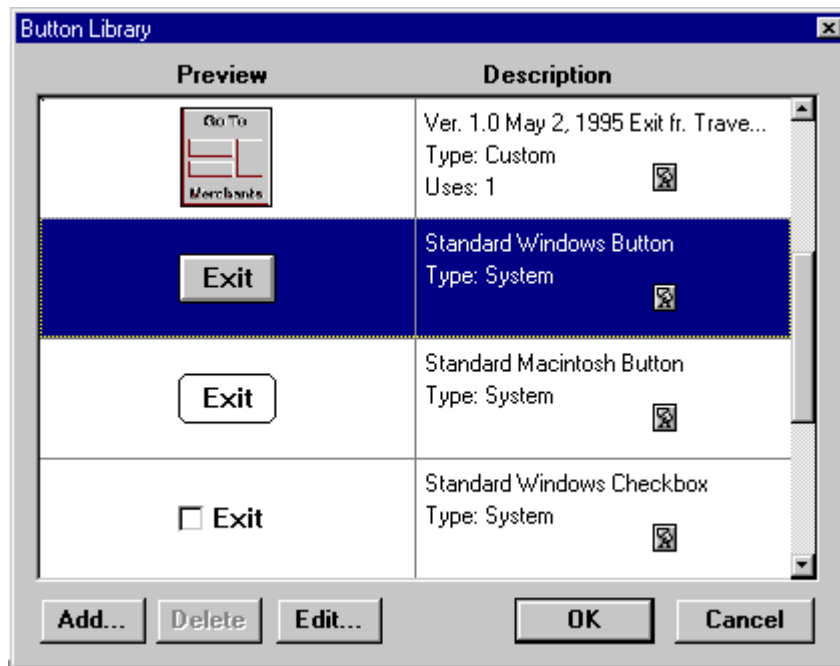
- Repeat the whole procedure for the **Next** button.
- The **Exit** button, is special in as much as we will want to change the image that is the default.
- First double click on the **Exit** button , and open the dialogue box:



- Now we will change the exit button from the default *large* one.
- Click in the **Button Type** box:



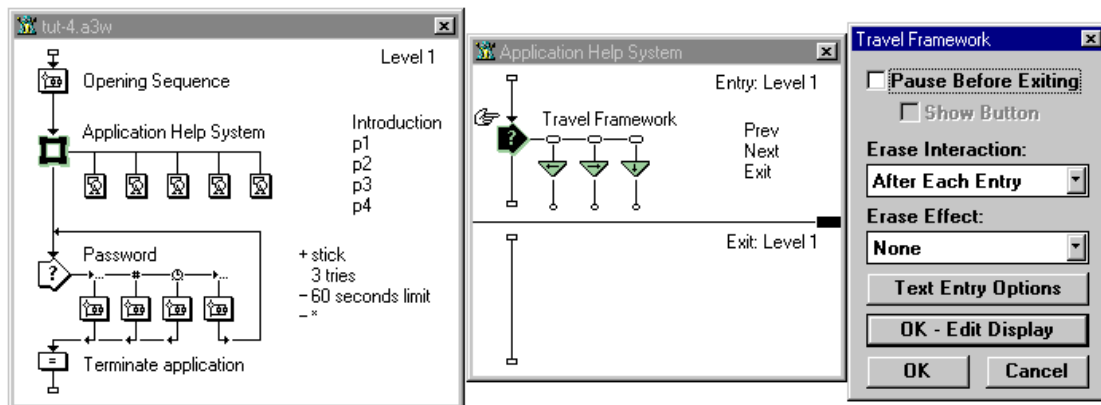
- Select the Standard Windows Button. You should now see this, then select **OK**



- Now, size and position the **Exit** button.

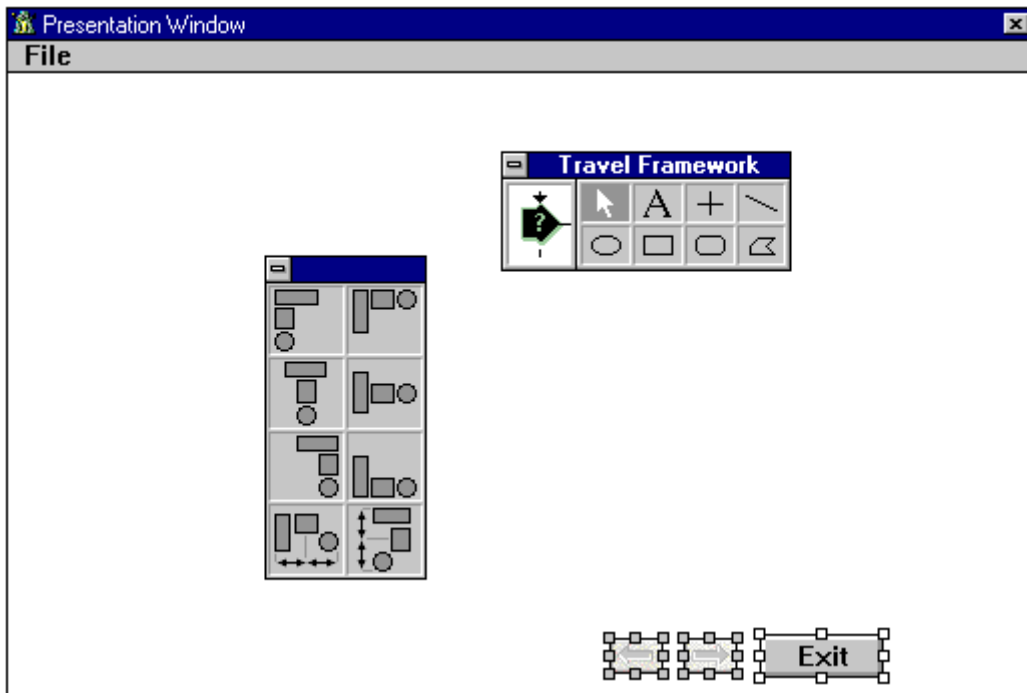
Of course, we don't know what all three buttons look like together, we now need to be able to exit their position in relationship to each other. The next few instructions will help us to do this (note. This procedure is very similar to that which we performed for the puzzle and the target areas).

- Double-click on the interaction icon contained within the framework icon, you will see the following dialogue box appear



- Click on the **OK - Edit Display** button

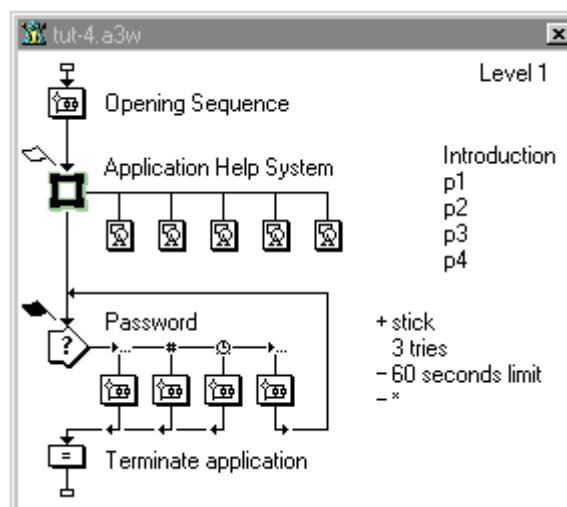
- You will see the familiar layout of the screen together with the three buttons, arrange them like shown below:



To help layout the object I made extensive use of the **Align Objects** tool, and in particular: (1) align objects horizontally; and (2) align objects spaced horizontally. Alignment is an important cue that end-users make use of when navigating - as such interface designers need to pay close attention to it.



We can now run our application help system, notice in the figure below I have added a start and stop flag around this section. This is just for speed and convenience. However, we are not yet finished, next we will add the flags to show that you have visited a particular page. Finally, the hypertext will be added.



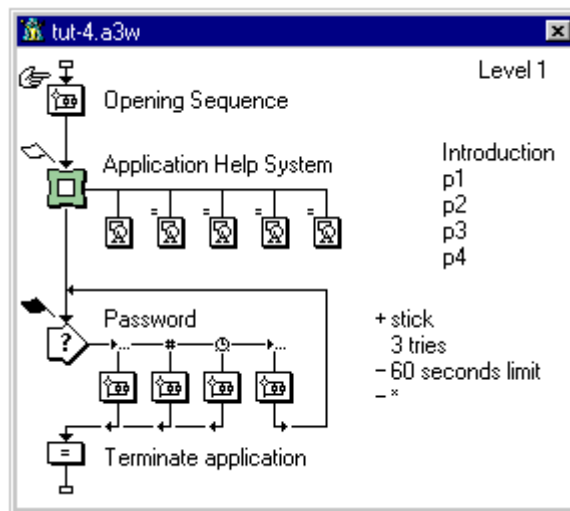
- Using CTRL+= add a calculation to each of the display icons (i.e. p1, p2, p3, and p4) and set the flags with appropriate values .

- Select the p1 display icon,
- Press CTRL+=, thus opening an empty calculation box
- Show Variables = Shift+CTRL+V and paste **one_seen**, type in:



- Repeat this process for the pages p2 through p4. Thus when someone uses the application help system they will now know which pages they have seen.

Your finished application should now be looking something like this:

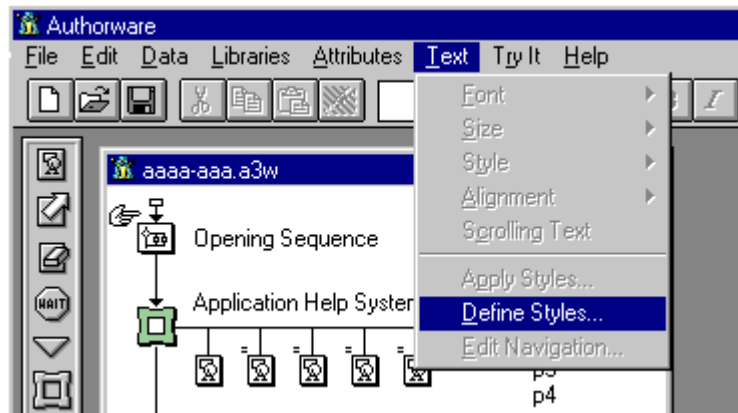


The following screen shows what the application help screen looks like after a end-user has worked through all of the slides and has returned to the main menu page.

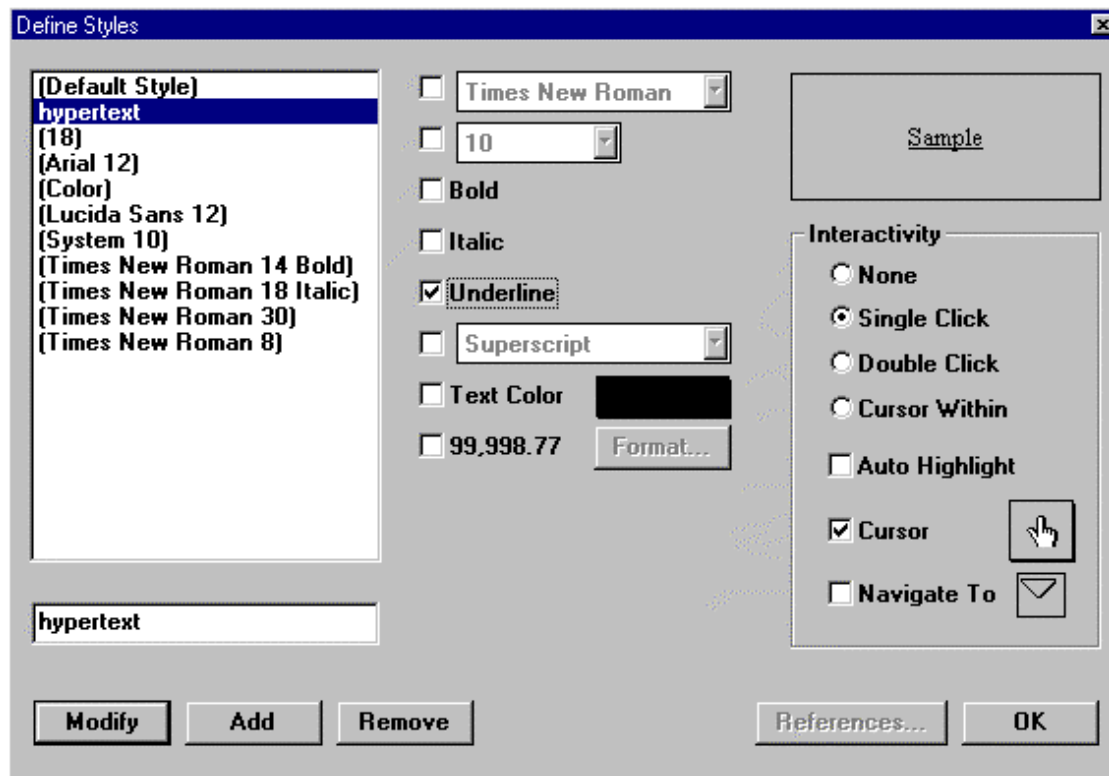


As a final touch we will now add the hypertext element to the application. Before we can actually add hypertext to the application we will need to define a hypertext text style.

- Go to the **Text** menu and select **Define Styles...** from the drop-down menu.



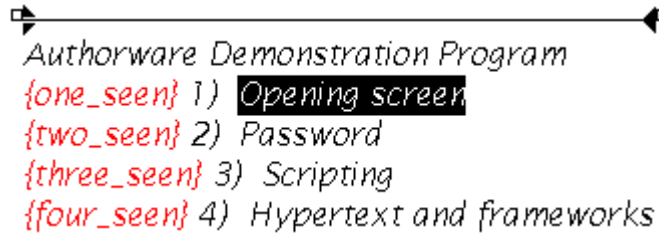
- Create a **hypertext** text style as described below, important points: (1) Underline signifies a hyper link; (2) cursor changes when over a hyperlink area, into a hand pointing, a user-friendly touch.



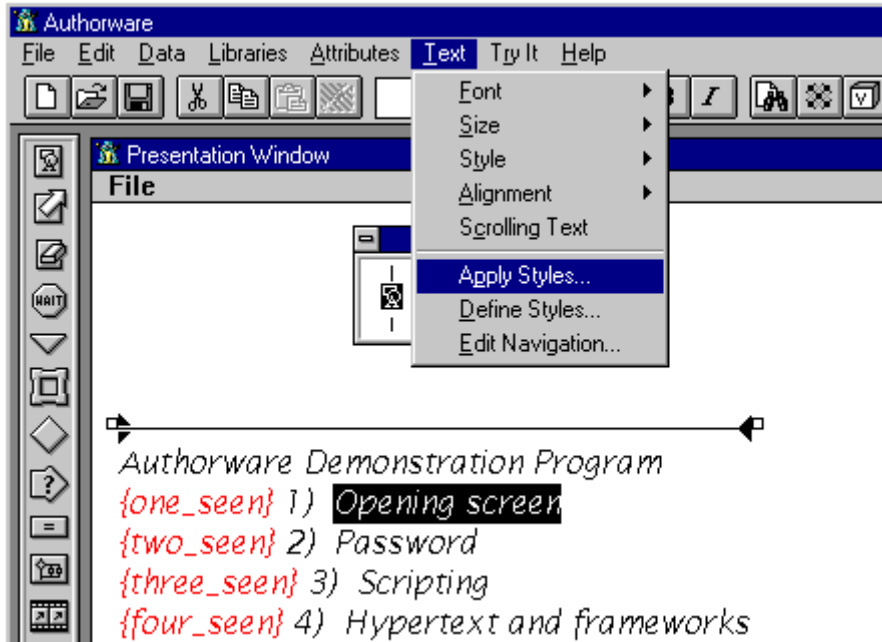
Having defined our hyperlink text style we can now apply it to any text within our application. I will show you how to do it for the first one - you can then go through the application on your own.

- First select a word that you which to make into a hyperlink, I will use the phrase "1) Opening screen" from the application help screen that we have just finished creating. I will link it to its related page (you can repeat this for each of the subsequent menu options listed).

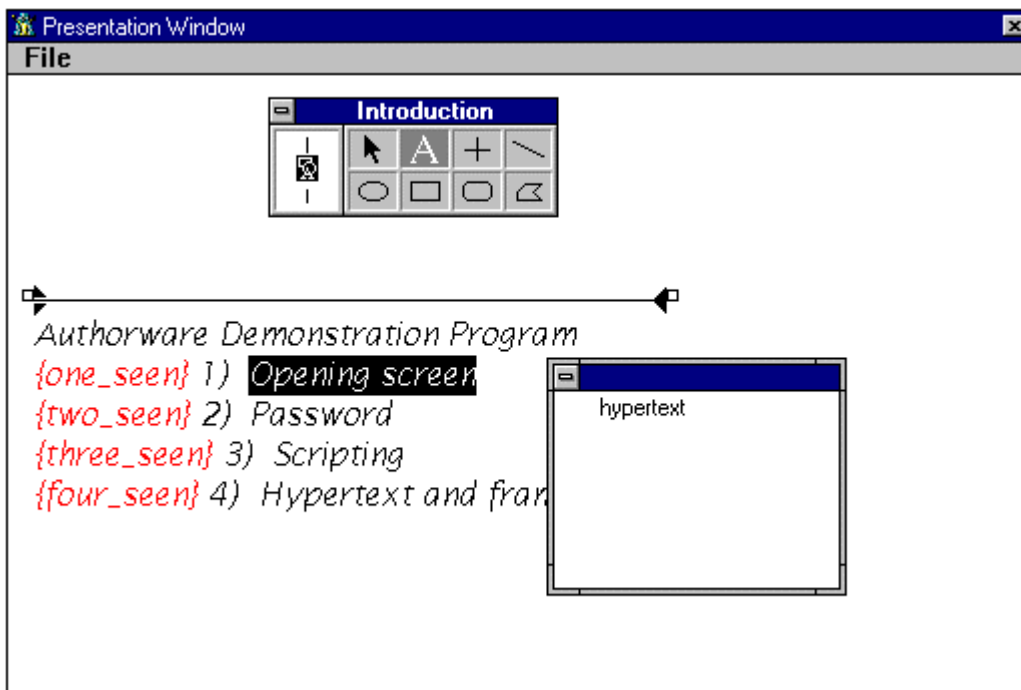
- First, highlight the word:


 Authorware Demonstration Program
 {one_seen} 1) **Opening screen**
 {two_seen} 2) Password
 {three_seen} 3) Scripting
 {four_seen} 4) Hypertext and frameworks

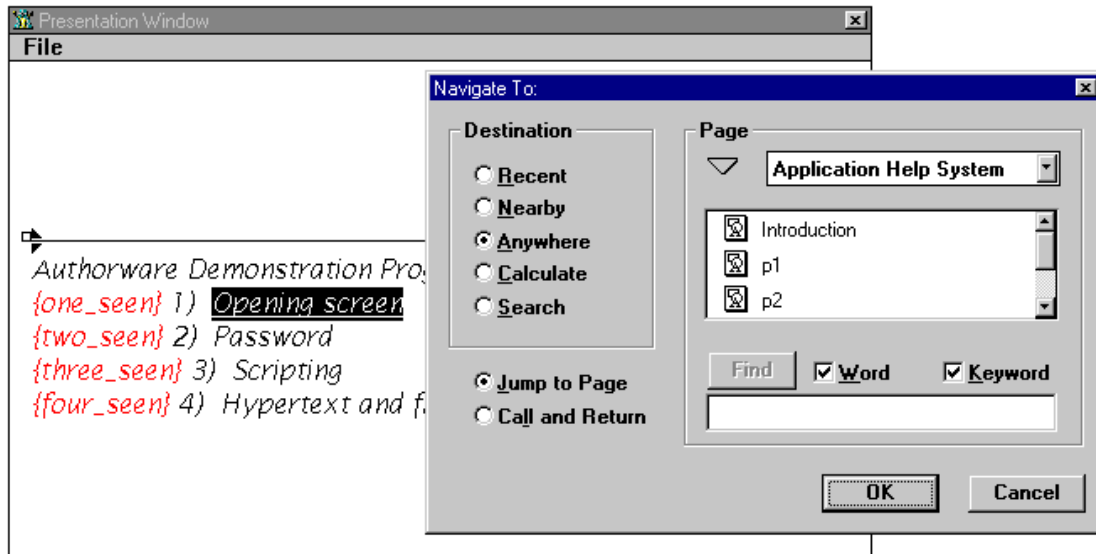
- You now define what you have highlighted as a hyperlink, to do this go to the **Text** menu, and select **Apply Styles**



- You will now see a selection of styles that can be applied, (in this case only one), select **hypertext**.



- You will now notice that the term “Opening screen” is underlined, as all hypertext will be as it is defined in the style. You now have to define the type of hypertext link - i.e. where the link should go to.



- Click on the page you want the link to jump to, in this case **p1**.
- Repeat this operation for the other three options.

We now have a fully functional application. Test it and ensure that it works how *you* think it should. If you have any questions about how it all fits together now would be a good time to put the questions down on paper and ask.

When selecting a flag to mark when something has happened, it is important to select an appropriate marker. I used a *, can you think of something better? Remember you can always use a different font to achieve the effect that you want, (e.g. the Picto font: **A B C D E F G H I J Z a b c d 1 2 3 4 5 6 7 * I**).

Summary

- CTRL+H = Align Objects
- CTRL+= = Can't stress enough how useful this command is! The facility to be able to attach a calculation to any icon is really useful.

In your own time

In General

- Examine the “File Setup” option, how many icons have you used (icons used so far: _____), you have a maximum of 50 with Authorware Working Model
- Notice how big the file is (current size: _____). Now save the file and compress it and go back to the “File Setup” menu and see how big the file is now (new compressed size: _____). Do you notice any difference? This is one of Authorware's more annoying features - you have to opt to compress the file, it is not done by default when you save it, so keep an eye on you file size.
- Play with the alignment tool set (i.e. display icons and CTRL+H)
- Define Text Styles... Play with this command and see what combinations you can come up with.
- Hypertext - this uses no extra icons
- CTRL+= uses no extra icons

Framework Icon

- Play with the “Effect Between Pages” option
- Add more buttons to the framework set - e.g. a jump to, search

5th Tutorial - Advanced features investigation

Features to be investigated: (a) file operations - writing; (b) questionnaires and multiple inputs; (c) multiple text entries...

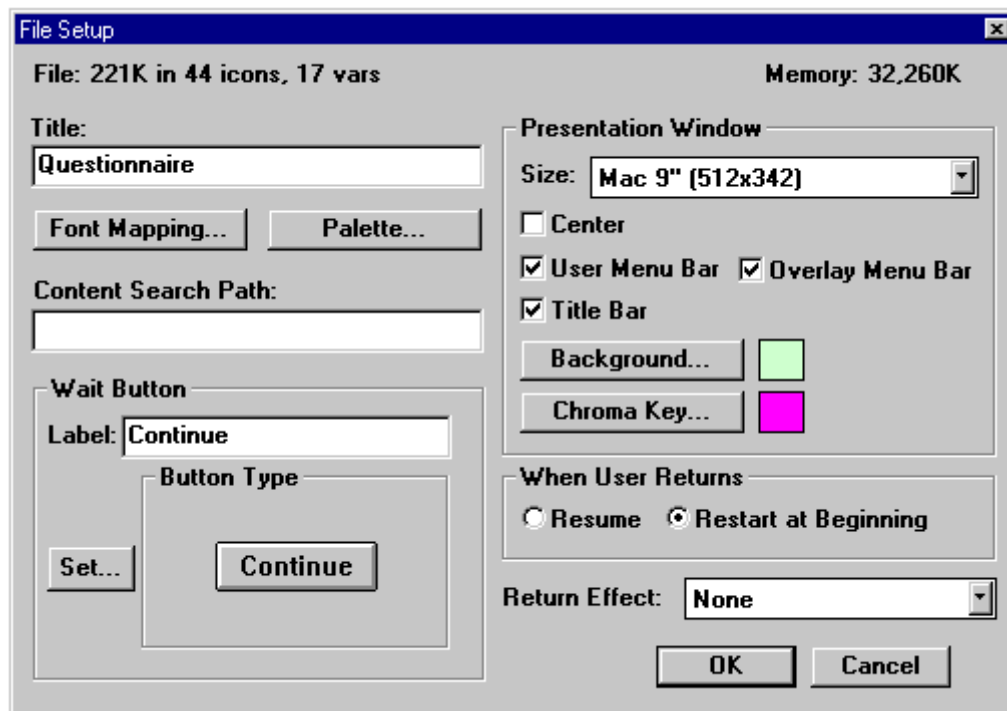
1) Towards more *intelligent* interactions, a questionnaire!

In this the penultimate tutorial, you will be creating a questionnaire. The aim is to build on what you have already learned (i.e. practising) and adding some new skills.

The objective of the tutorial is to create a questionnaire which asks a number of questions and then stores them to a file. The questions are divided into two parts: (1) personal details; and (2) actual questions.

- | | |
|-------------------|--|
| Personal Question | <ul style="list-style-type: none">• Name• Age• Sex |
| Actual Question | <ul style="list-style-type: none">• How would you rate your computer literacy?• How would you rate your acceptance of and general attitude towards new technologies?• I am enjoying the end-user interface module. |

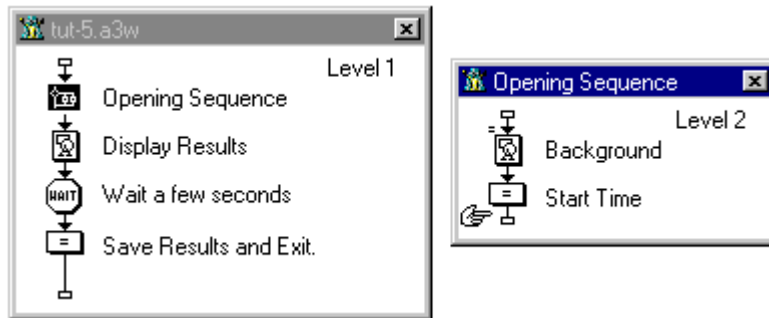
Configure the application like:



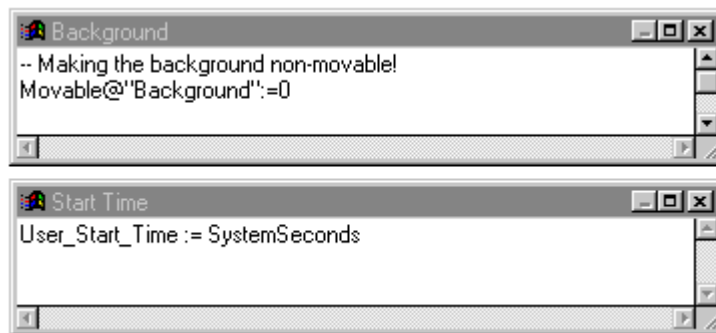
From the above dialogue box, taken from the finished application, notice how many icons and variables have been used! Also the size of the finished application.

- To begin configure the application as shown above.

- Next add the appropriate icons to the flow line, as shown in the below dialogue box. We will then work through them and add their details.

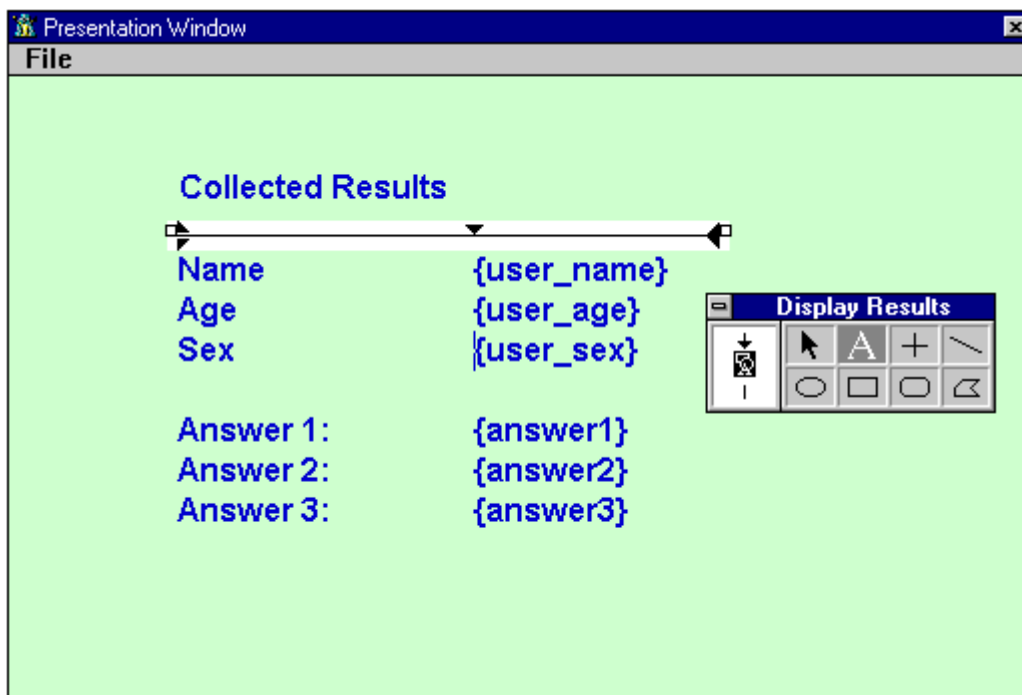


- Two calculations are in the **Opening Sequence** group. Add and configure as appropriately.



The first command **movable** configures an object to be unmovable at run time (CTRL+P still works though). The second command records the start time of the application, we will use this variable later to calculate the total time taken by the end-user in the application.

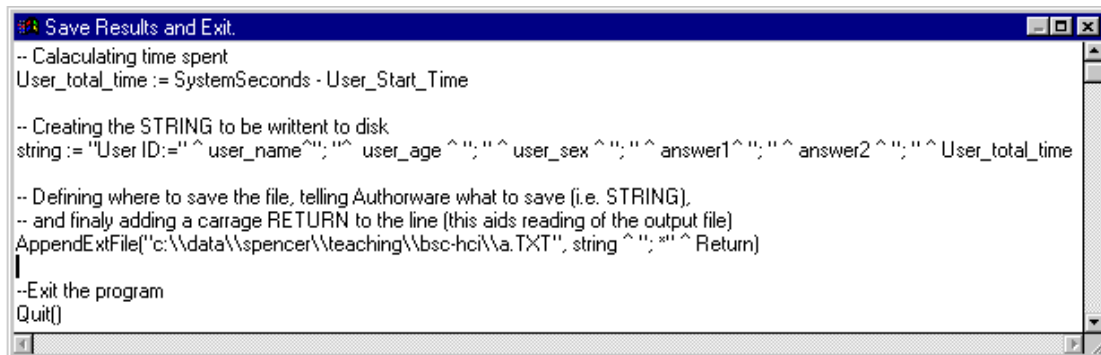
- Put any background in the **Background** display icon!
- Configure the **Display Results** icon as shown below:



- When asked for the variable details enter **0** as the **Initial Value**.
- Configure the **Wait** icon as show:
- Position the timer later, using the pause (i.e. CTRL+P) at run time.



- Now enter in the following calculations into the **Save Results and Exit** calculation icon.



The first calculation works out the **User_total_time**, this is the total time taken by the end-user to work through the application. Next a **string** is created, this is then written to a text file. When writing to a file the keyword **Return** is used to create a new line. The **^** are used to link text and variables entries together.

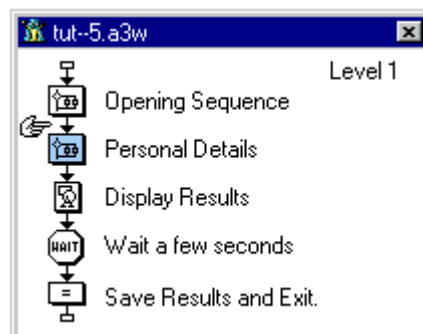
```
User ID:=spencer ; 25; male; 4; 4; 24.215999999999999; *
User ID:=spencer; 25; male; 2; 3; 29.159000000000001; *
```

Table 1 Example Data

The above is an example of a set of record written to the text file. Notice how each record appears on a separate line (i.e. **Return**), and how the time is formatted. Information of this sort can be easily imported into either a spreadsheet or another program.

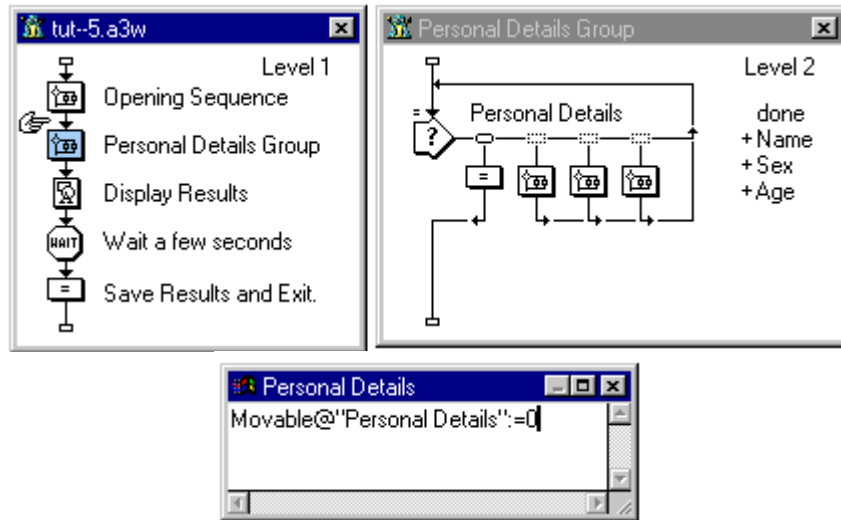
Now the hard work begins... Creating the personal details entry section.

- Add a group icon to the flow line, just below **Opening Sequence** group icon
- Call this new group icon **Personal Details**

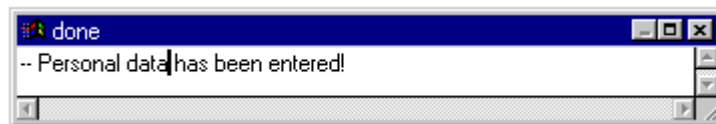


Authorware only permits one **EntryText** at a time. As such we can not have multiple text entry boxes on the screen. What you will be shown next is a *work around*. In essence we have three **Hot Spots**, what when click on allow the end-user to enter in their details, one at a time.

- First, we place a **Interaction** icon on the flow line.
- Then we add a **Button** and three **Hot Spots**, the flow line should look like this:

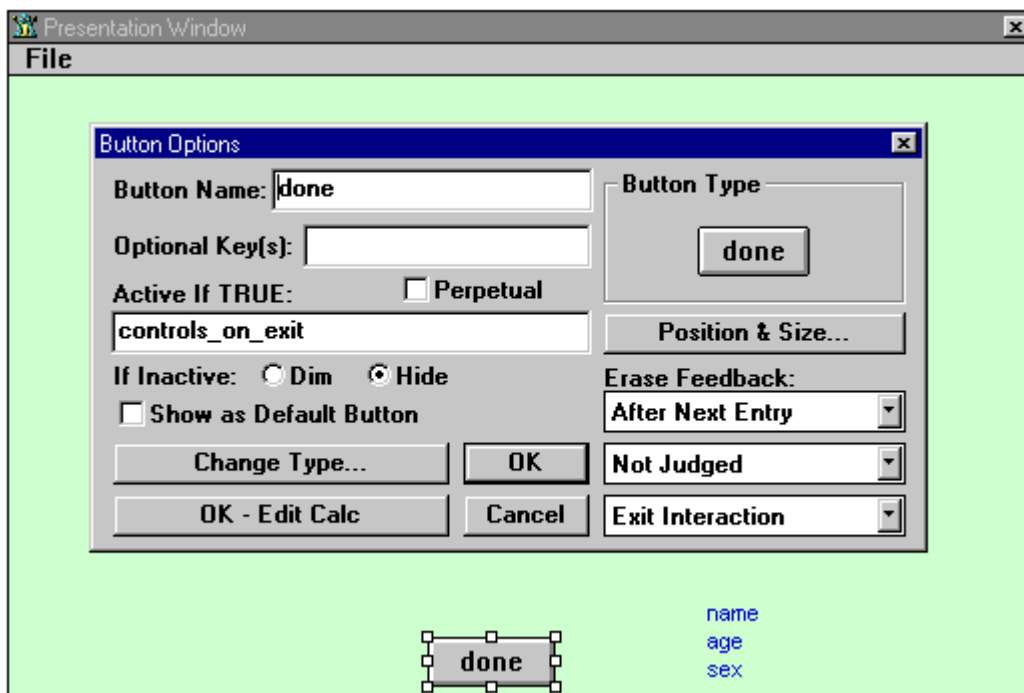


The **done** button has a calculation icon beneath it, all it contains is a comment:



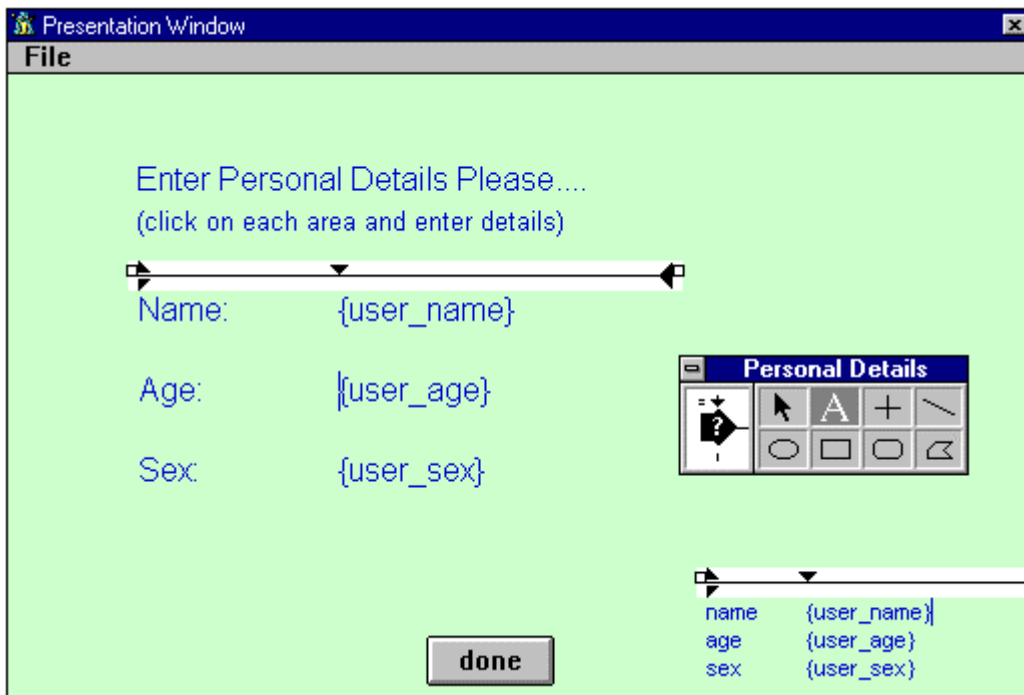
I will now build this up a bit at a time:

- First things first, the **done** button, configure like shown below:



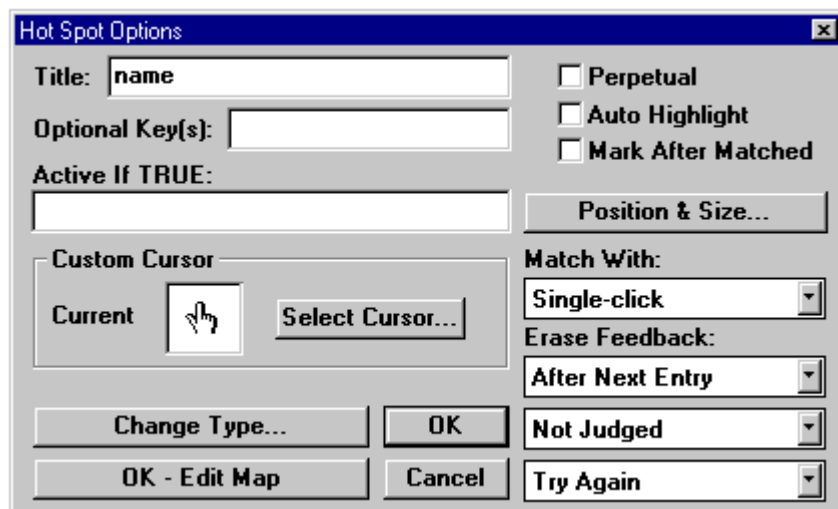
- The variable **controls_on_exit** defaults value should be **FALSE**. Also, don't forget the **If Inactive = Hide**, this directly linked to **controls_on_exit**. When the **controls_on_exit** variable is **TRUE** the button will be shown, else it will be hidden.

- Next we will set up the interaction icon's display

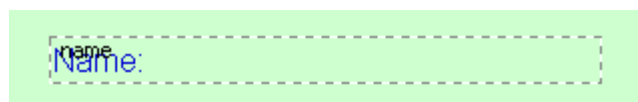


There are two copies of the variables being displayed. The main variable display is in the middle of the screen. On the bottom right is a smaller copy of the variables being displayed - this is for debugging purposes.

- Configuring the **Name** hot spot, the dialogue box should look like this:

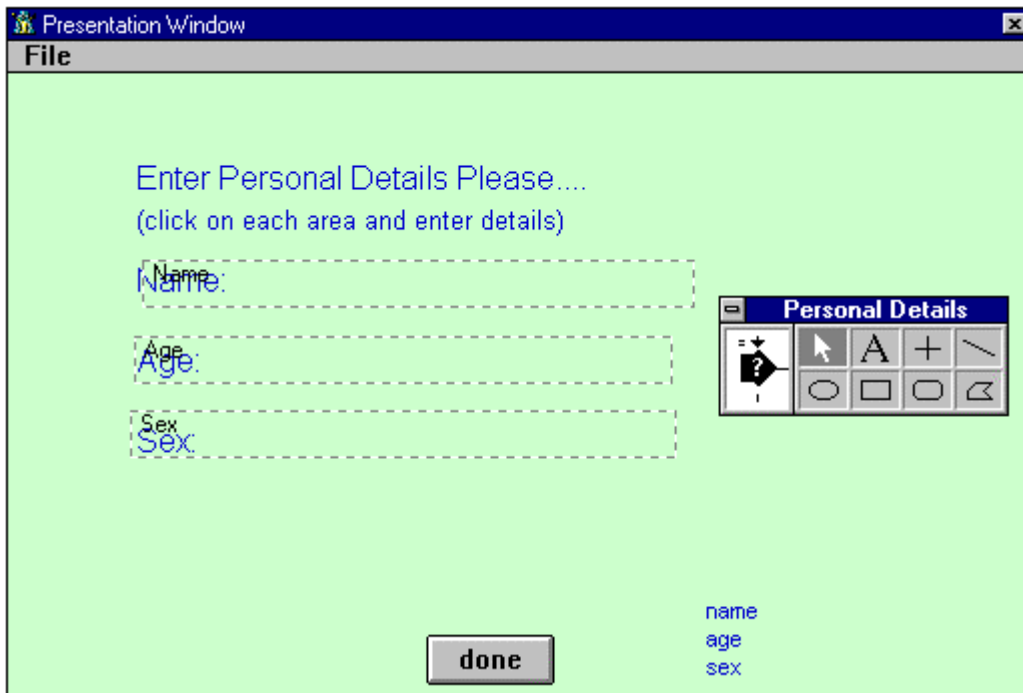


- To get to this point you will have to alter: **Custom Cursor** and **Title**.
- The **Hot Spot** for **Name** should be all the way across "Name {user_name}", like:



- Repeat this for **Age** and **Sex**.

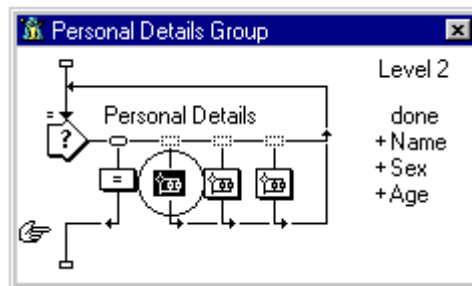
- The **Interaction** dialogue box will look like:



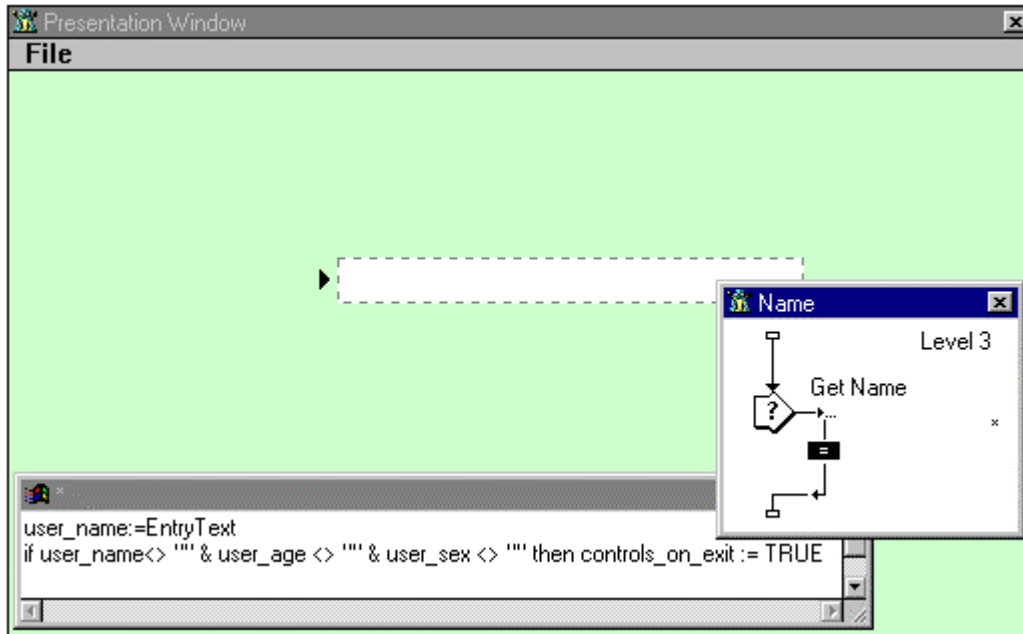
The screen shot above, shows clearly the **Hot Spots** in relation to the **Interaction** icon's variables being displayed (i.e. **name** {**user_name**}, **age** {**user_age**}, and **sex** {**sex**}).

Now we will get the individual personal details. To get this to work we have to marry the **Hot Spots** and the **EntryText** areas together. That is, when a end-user clicks on **Name** they then enter in their name in roughly the same spot.

- First open the **Name** group:



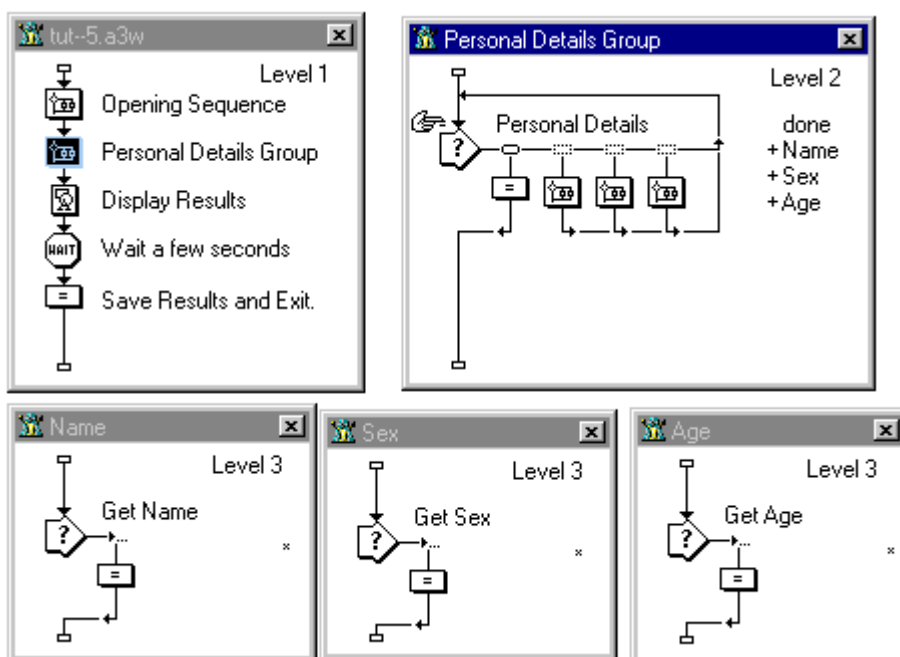
- Now add an **Interaction** icon
- Next add a calculation icon to the **Interaction**, use a **TextEntry** response type.
- As discussed above the idea is to have the **TextEntry** area in the same position as where the responses are displayed.
- Position the **TextEntry** area as shown below:



- Enter the details as shown above into the **Calculation** icon, which is attached to the **TextEntry**.
- Repeat this process for the other two personal details.

Of course, there are differences between the three interactions. They are: (1) the position of the TextEntry areas, these have however been shown on the previous page; (2) the calculation "user_XXXX:=EntryText" will differ for each of the three.

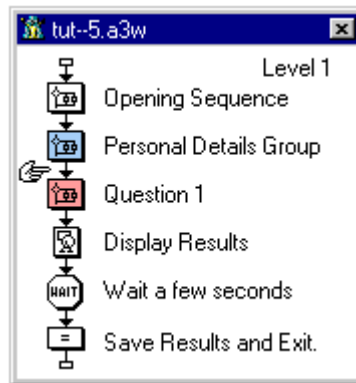
The application should look like this now:



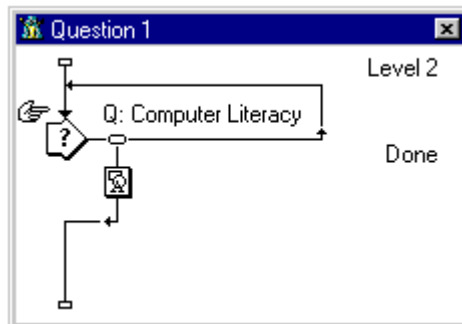
Even though the application is not yet finished you should still be able to run the application. When you run the application you will notice the effect of the **Hot Spots** and the **TextEntry** areas. If the two areas don't match up use the pause facility (CTRL+P) to move things around. Also, check that the results are saved to the file in the correct format (see Table 1 Example Data).

Having obtained the end-users personal details, next the questions are to be displayed and their responses stored. This stage of the questionnaire is quite quick as once the first question has been configured only minimal changes are required to create the next, (i.e. use cut and paste, followed by minimal editing).

- First add another group icon to the flow line, as shown:

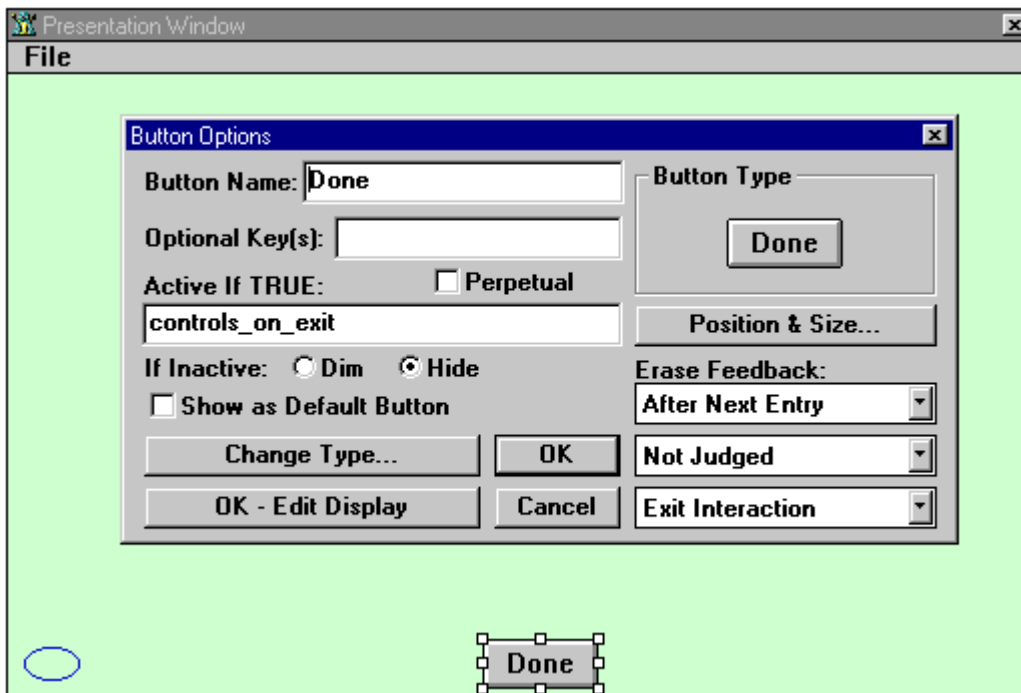


- Open the group icon and drag a **Interaction** icon to the flow line:



The display icon attached to the **Done** button it a "empty", just put something rubbish in it.

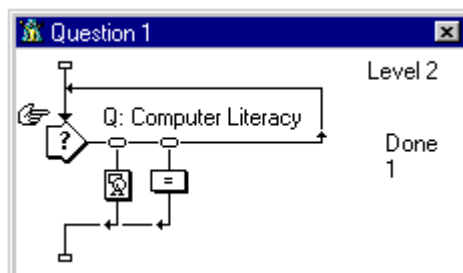
- Configure the **Done** button as shown below:



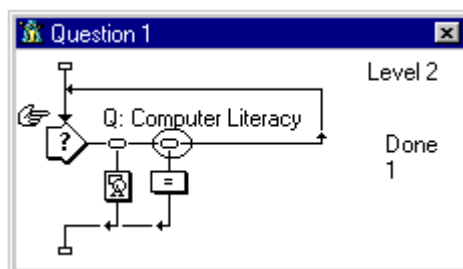
We are again making use of the variable **controls_on_exit**.

Next we will add and position the five **Radio Buttons**. End-users will have the ability to select only one of the five buttons. The selection is recorded in the variable **answer1**. The answer is later save to the text file.

- Drag a calculation icon to the flow line, next to the **Done** button, label the calculation 1.

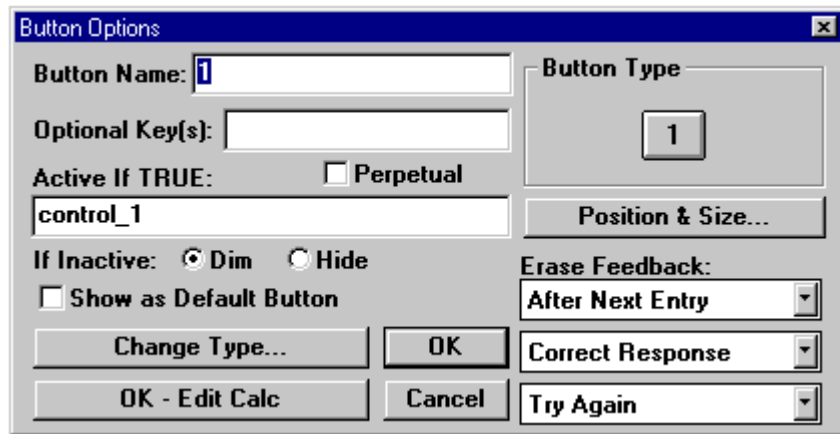


- Now double-click on the **Response Type** button, as indicated by the circle:

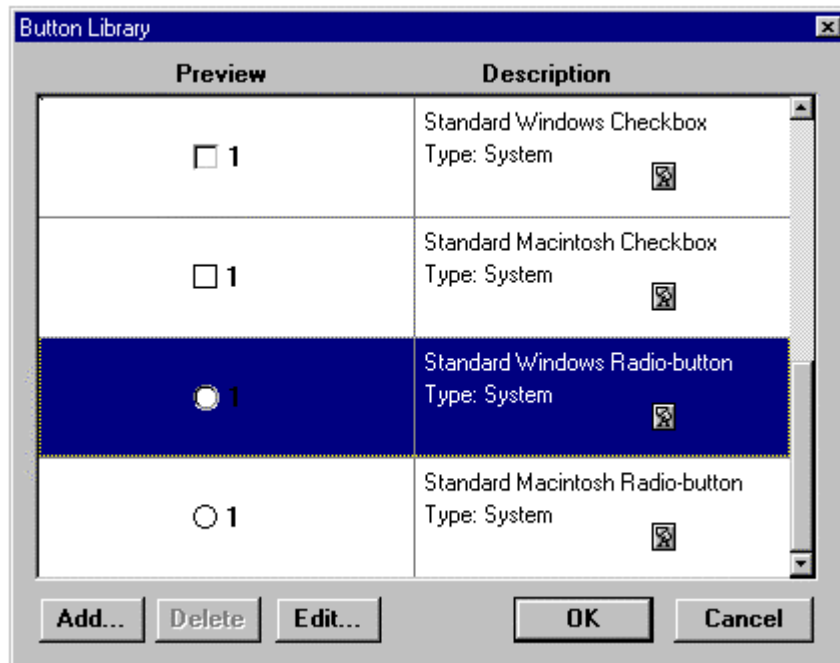


Now we will configure the button to be a Radio Button and Position it appropriately.

- Configure the Button dialogue box to look like

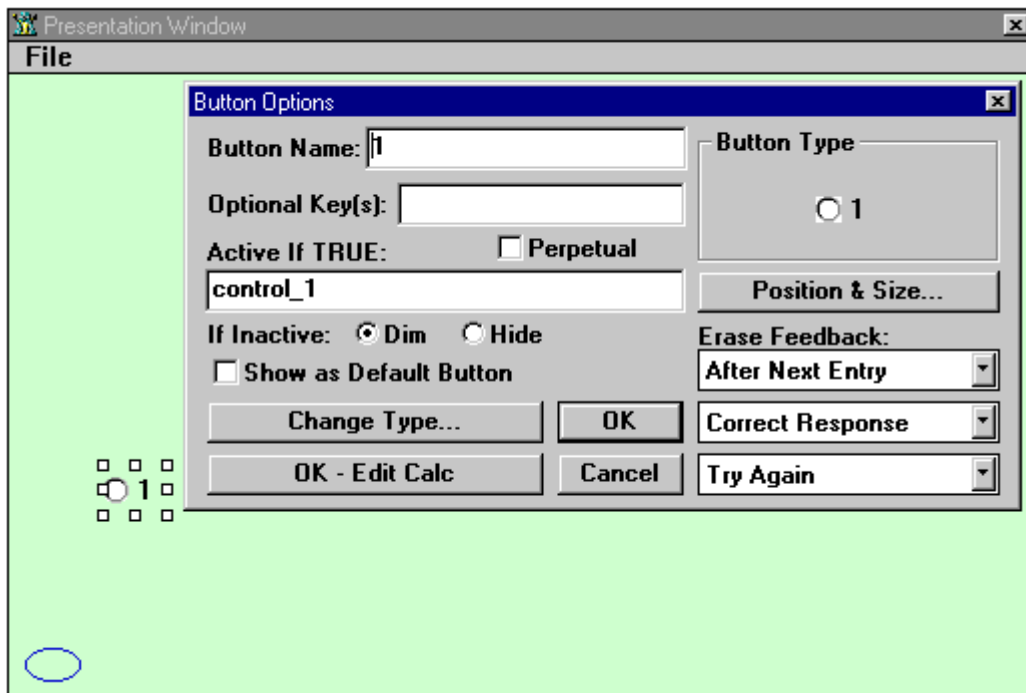


- Click on the **Button Type**
- You will now see the following dialogue box:

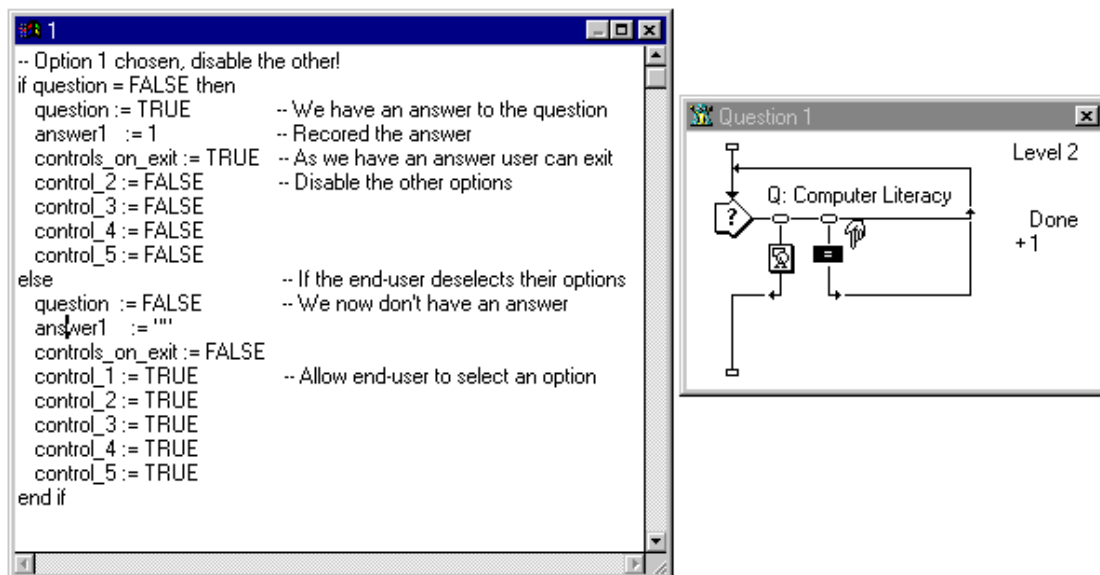


- Select the **Radio-button**, as indicated above

- You dialogue box, and screen (position the **radio-button** as shown) should now look something like this:



- Next enter the following details in the calculation icon:



- Repeat this calculation (with appropriate edits) for the other four **Radio-buttons** (details of calculations with edits made are shown on the next page). Use cut and paste (CTRL+C and CTRL+V respectively) to copy calculations, rather than re-type each one. This will save you a lot of time, and effort debugging

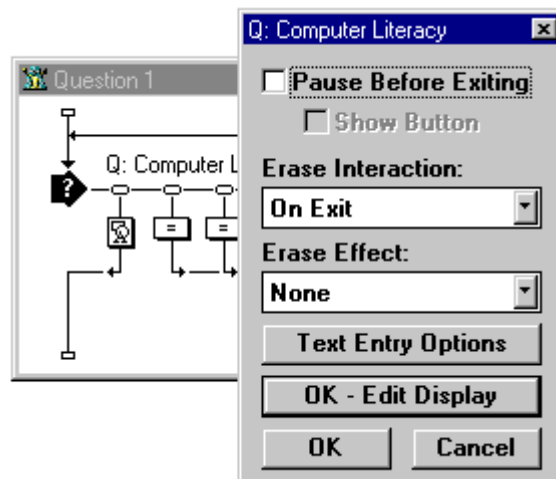
Of course, the calculations are no the same for all the calculations. The table below shows what entries should be made for each of the other four calculation icons. Bold entries are the ones to watch, (i.e. check that you have entered them correctly into the different question groups).

| | | | |
|--|--|--|--|
| <pre>-- Option 2 chosen if question = FALSE then question := TRUE answer1 := 2 controls_on_exit := TRUE control_1 := FALSE control_3 := FALSE control_4 := FALSE control_5 := FALSE else question := FALSE answer1 := "" controls_on_exit := FALSE control_1 := TRUE control_2 := TRUE control_3 := TRUE control_4 := TRUE control_5 := TRUE end if</pre> | <pre>-- Option 3 chosen if question = FALSE then question := TRUE answer1 := 3 controls_on_exit := TRUE control_1 := FALSE control_2 := FALSE control_4 := FALSE control_5 := FALSE else question := FALSE answer1 := "" controls_on_exit := FALSE control_1 := TRUE control_2 := TRUE control_3 := TRUE control_4 := TRUE control_5 := TRUE end if</pre> | <pre>-- Option 4 chosen if question = FALSE then question := TRUE answer1 := 4 controls_on_exit := TRUE control_1 := FALSE control_2 := FALSE control_3 := FALSE control_5 := FALSE else question := FALSE answer1 := "" controls_on_exit := FALSE control_1 := TRUE control_2 := TRUE control_3 := TRUE control_4 := TRUE control_5 := TRUE end if</pre> | <pre>-- Option 5 chosen if question = FALSE then question := TRUE answer1 := 5 controls_on_exit := TRUE control_1 := FALSE control_2 := FALSE control_3 := FALSE control_4 := FALSE else question := FALSE answer1 := "" controls_on_exit := FALSE control_1 := TRUE control_2 := TRUE control_3 := TRUE control_4 := TRUE control_5 := TRUE end if</pre> |
|--|--|--|--|

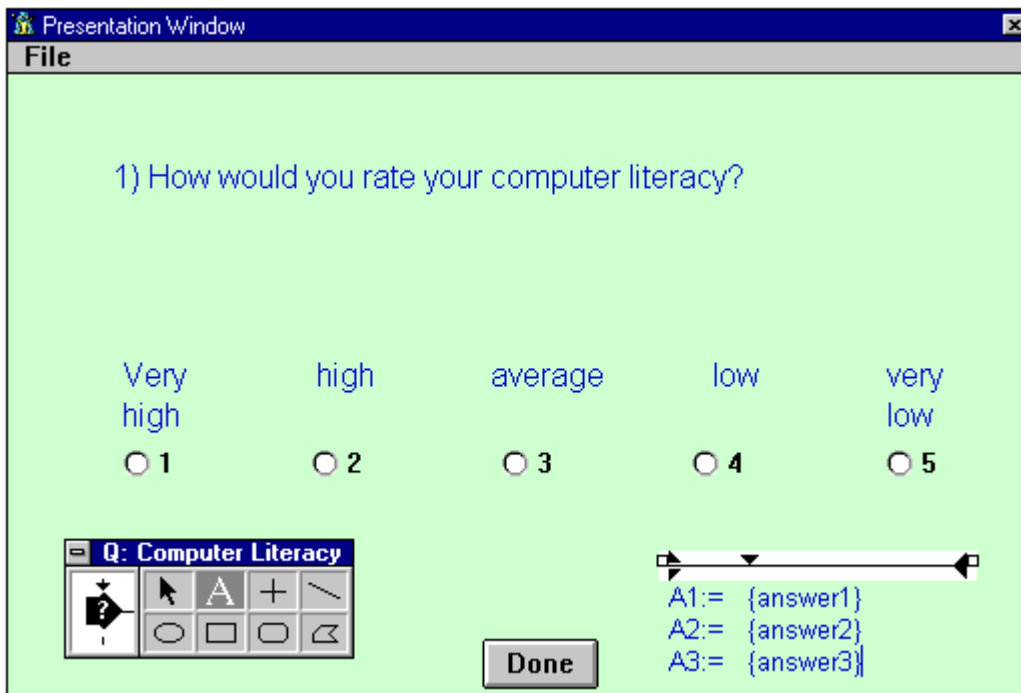
In essence, all that is happening is: (1) the first if is entered when the end-user selects an option. As such it is recorded (into the **answer1** variable) and the other radio-buttons are disabled; (2) if the end-user subsequently decided that their original choice was incorrect **else** takes care of it. It enables all of the radio-buttons and resets the answer variable.

If this seems a little confusing save two version of the program one with all of the calculations and one without and see what the differences are. Alternatively ask for some help.

- Finally, edit the **Interaction** icon, that is, double-click on the interaction icon
- Select **Ok - Edit Display**



- Edit the **Interaction** display screen to look like this:

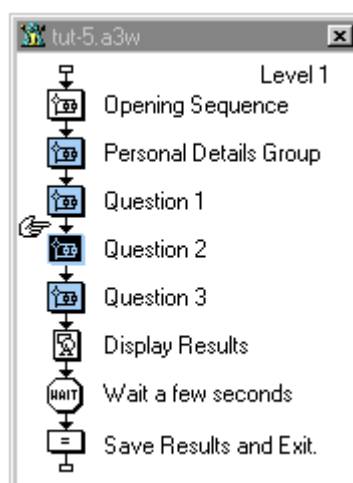


- Useful tools for editing the screen are: Alignment (CTRL+H) and Modes (CTRL+M).

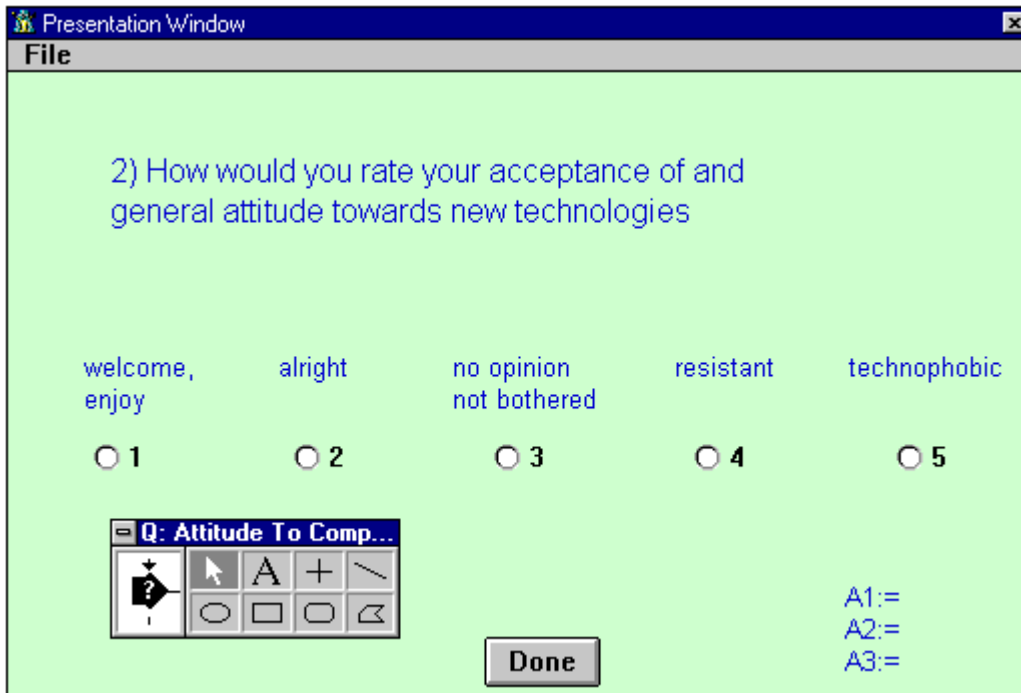
Much as with the personal details, on the bottom right of the screen there is a debugging variable display box. This is particularly useful for checking that you record the correct entries and for checking that you don't overwrite any variables.

You can now run the program. This time you will notice that more information is written to the text file. Of course, we are not yet finished two more questions are to be added. This is now very simple. All we need to do is copy and paste **Question 1** to create **Question 2** and **Question 3**.

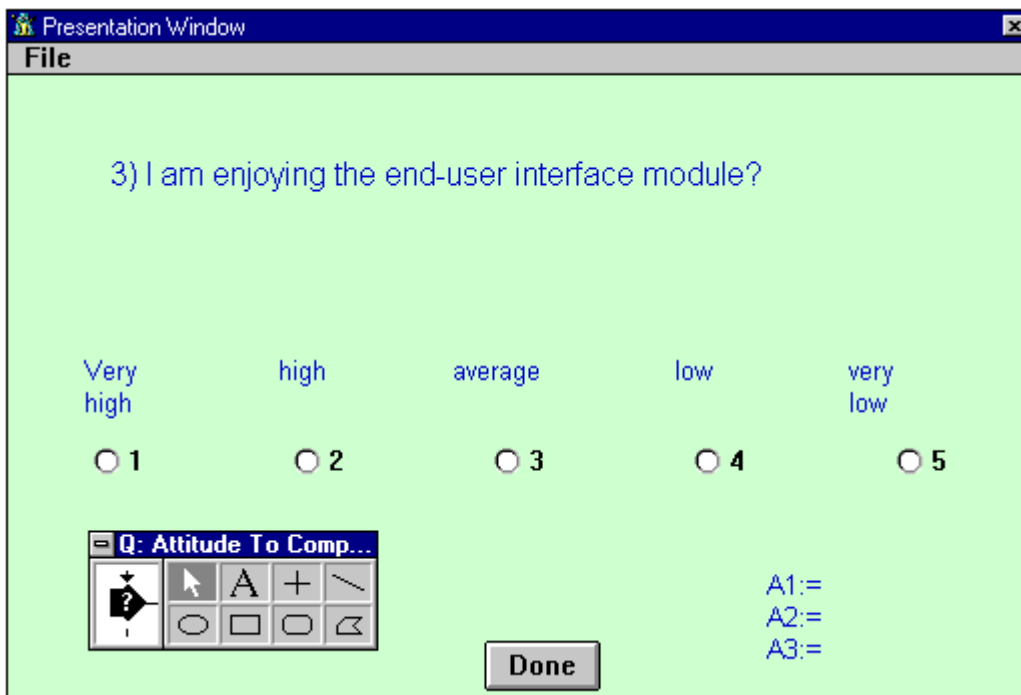
- Copy and paste **Question 1** to create **Question 2** and **Question 3**



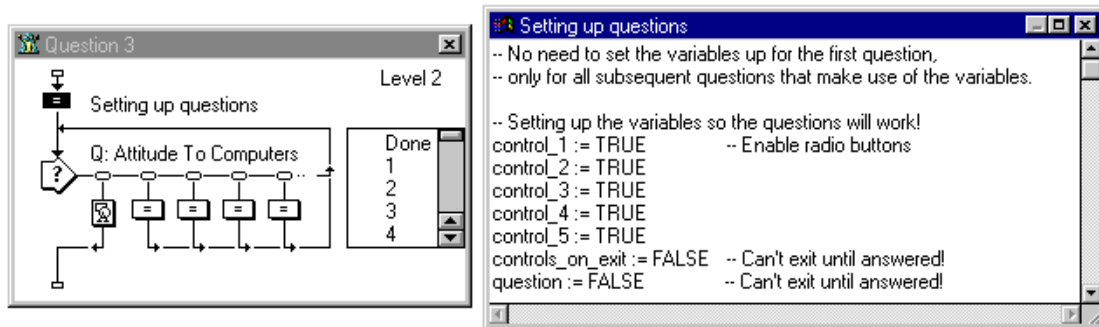
- Special amendments to be made to **Question 2** and **Question 3** Step by step
- First alter the **Interaction** display screen to contain the correct questions:



...



- Next, the calculations: Add a calculation icon to the start of both **Question 2** and **Question 3**. This sets up the variables so that they can be reused.



- Examples of amendments to calculations for Question 2, items in bold are the changes. Similar changes are made for the third question, (i.e. **answer3** as opposed to **answer1** or **answer2**).

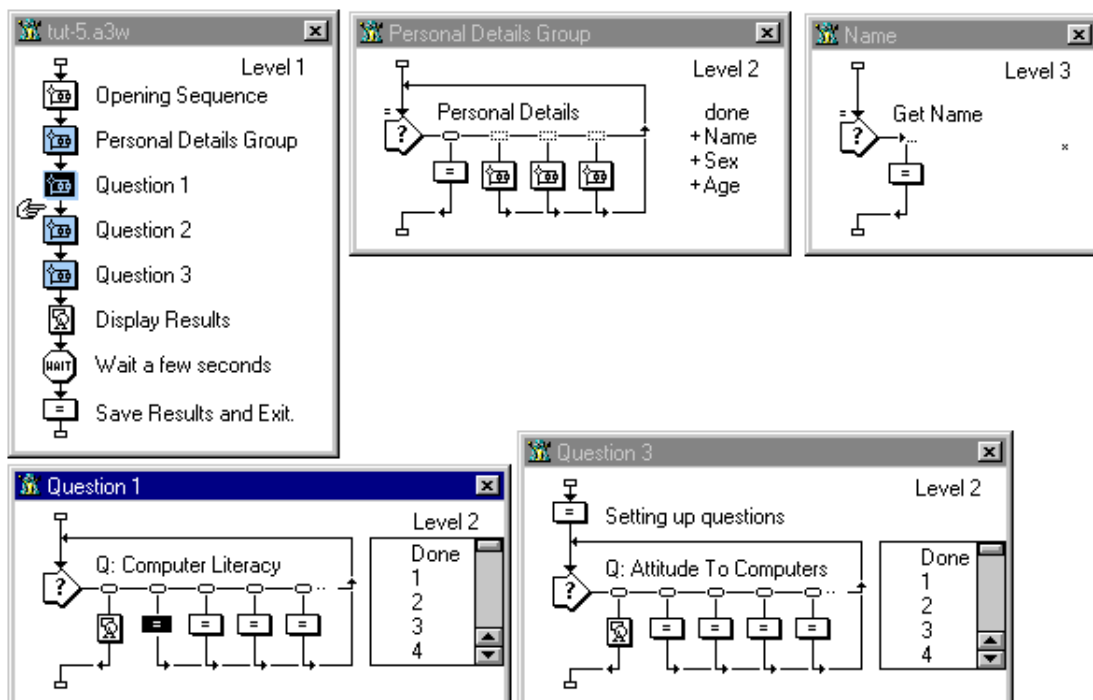
```
-- Option 2 chosen
if question = FALSE then
  question := TRUE
  answer2 := 2
  controls_on_exit := TRUE
  control_1 := FALSE
  control_3 := FALSE
  control_4 := FALSE
  control_5 := FALSE
else
  question := FALSE
  answer2 := ""
  controls_on_exit := FALSE
  control_1 := TRUE
  control_2 := TRUE
  control_3 := TRUE
  control_4 := TRUE
  control_5 := TRUE
end if
```

```
-- Option 3 chosen
if question = FALSE then
  question := TRUE
  answer2 := 3
  controls_on_exit := TRUE
  control_1 := FALSE
  control_2 := FALSE
  control_4 := FALSE
  control_5 := FALSE
else
  question := FALSE
  answer2 := ""
  controls_on_exit := FALSE
  control_1 := TRUE
  control_2 := TRUE
  control_3 := TRUE
  control_4 := TRUE
  control_5 := TRUE
end if
```

```
-- Option 4 chosen
if question = FALSE then
  question := TRUE
  answer2 := 4
  controls_on_exit := TRUE
  control_1 := FALSE
  control_2 := FALSE
  control_3 := FALSE
  control_5 := FALSE
else
  question := FALSE
  answer2 := ""
  controls_on_exit := FALSE
  control_1 := TRUE
  control_2 := TRUE
  control_3 := TRUE
  control_4 := TRUE
  control_5 := TRUE
end if
```

```
-- Option 5 chosen
if question = FALSE then
  question := TRUE
  answer2 := 5
  controls_on_exit := TRUE
  control_1 := FALSE
  control_2 := FALSE
  control_3 := FALSE
  control_4 := FALSE
else
  question := FALSE
  answer2 := ""
  controls_on_exit := FALSE
  control_1 := TRUE
  control_2 := TRUE
  control_3 := TRUE
  control_4 := TRUE
  control_5 := TRUE
end if
```

This is what your application should look like now, (this is a small selection):



Run the application now, to check that it performs how you would expect. It is important that you check that the program works - and you understand what is happening.

There are two errors in this program - clue one is in a script, the other is a missing script. Run the program a few times and see if you can spot it, and more importantly correct it.

Summary

- CTRL+H = Align Objects
- CTRL+M = Mode
- CTRL+= = This is still a useful command.
- CTRL+X and CTRL+V = The standard windows 3.x Cut and Paste commands

In your own time

In General

- Examine the “File Setup” option, how many icons have you used (icons used so far: _____), you have a maximum of 50 with Authorware Working Model
- Notice how big the file is (current size: _____). Now save the file and compress it and go back to the “File Setup” menu and see how big the file is now (new compressed size: _____). Do you notice any difference? This is one of Authorware’s more annoying features - you have to opt to compress the file, it is not done by default when you save it, so keep an eye on you file size.

Calculation Icon

- **Movable@”stick”=0** this makes a display object unmovable, especially useful for graphics. Indeed you might have noticed that it is essential - as your graphics keep being moved around when you click in the wrong place.
- **string** experiment with outputting details to a file. Also see how the results can be imported into a spreadsheet for analysis (e.g. MS Excel).

Interaction Icon

- **Hot Spot** ~ play with the different options that are available, for example: double-click, different cursors, auto highlight, mark after matched.
- **Radio-button** ~ Edit the default radio button and see what effects you can create. For example, specify that a sound is to be played when the users mouse is above the button.

6th Tutorial - Summary "What now?"

Authorware is on the surface a very simple application. As such, I am sure you now have an appreciation for the types of things that you can achieve with it. This tutorial is intended to take what you have learned so far and show how the extended features of Authorware can make further enhancements.

In this tutorial we will be *enhancing* the application that we created in tutorial five (i.e. the questionnaire). As you may remember there were a number of areas within which the application was not as user-friendly as it could have been. Below are a list of the areas of Authorware that we will be examining:

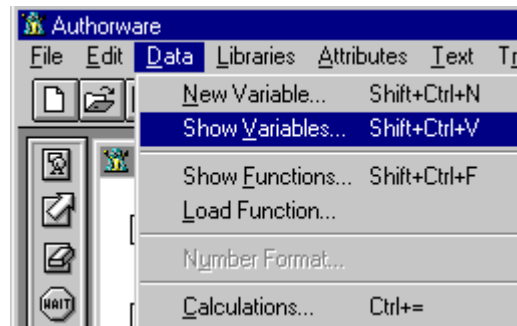
- Show variables/functions (paste command, PresetEntry, Preload)
- Defining tabs
- Defining fonts
- Defining buttons (Up/Down/Over/Disabled)

1) Variables/Functions

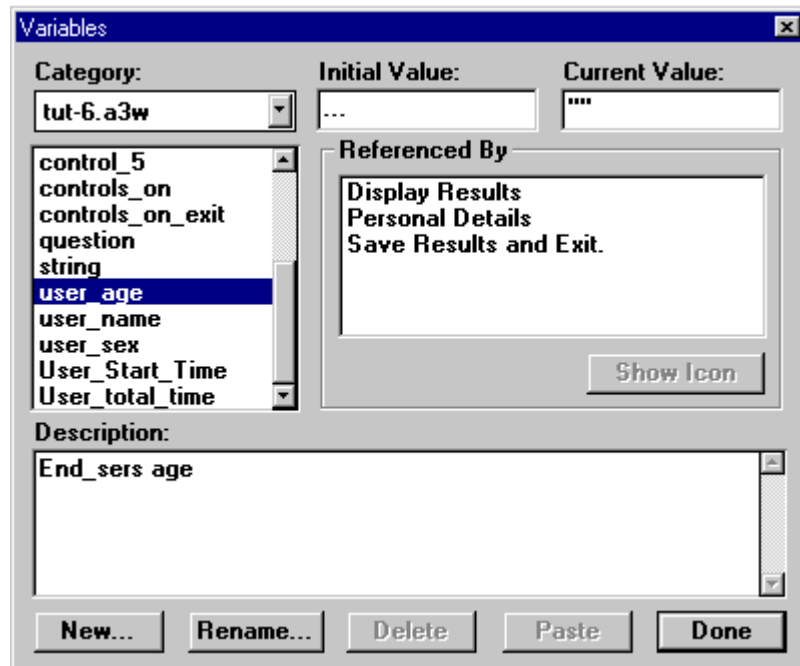
During our use of Authorware we have made use of both variables and functions. However, we have not as yet explored how to best use and access these facilities. To demonstrate this I will now introduce two new commands and one new procedure.

Editing and changing a variable. In tutorial five we asked the user to enter three. However, when running the program the variables were not as visible as they could have been. To solve this, rather than leave the variables blank we will now fill them with dots.

- Go to the menu, and select **Show Variables**

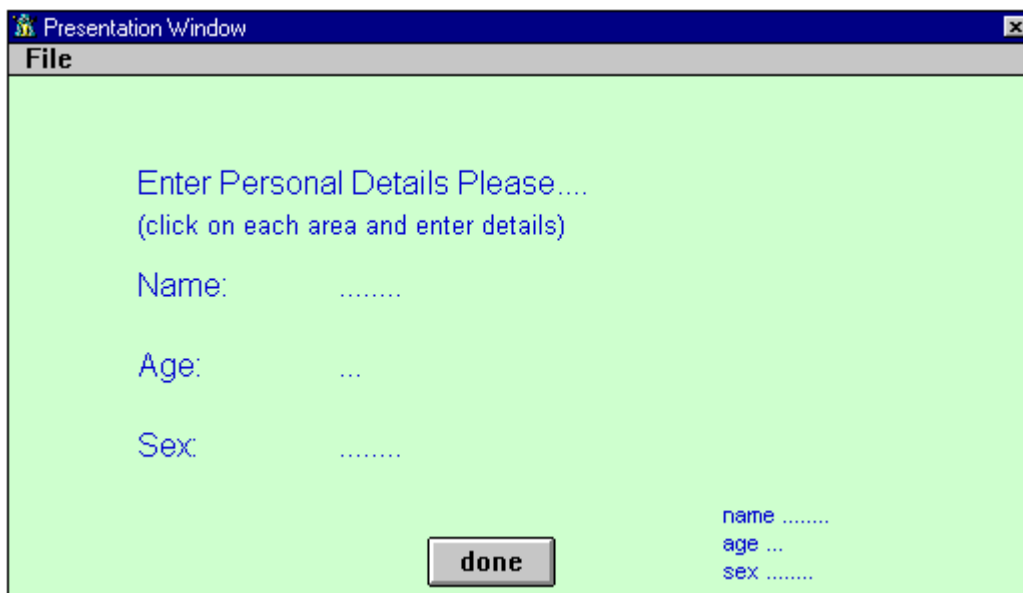


- You should now see the following dialogue box:



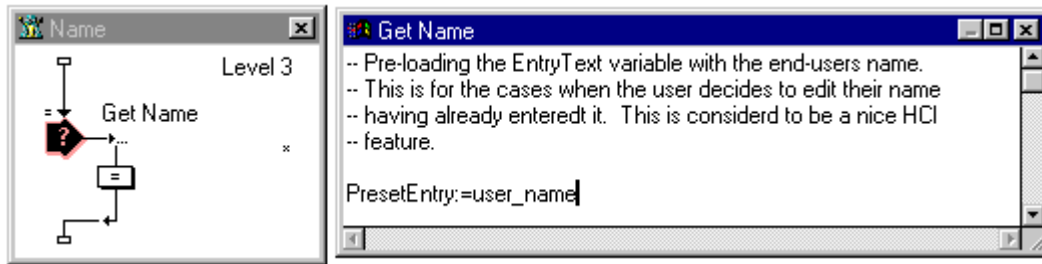
- As in above, select the name of your application from the **Category** box (top left).
- Then select the **user_age** variable and define the initial value to be ... (a series of dots).
- Repeat this process for your other two personal details variables, (i.e. **user_name** and **user_sex**)

When the program runs now the variables will be much more visible. The dots make the empty variables stand out.

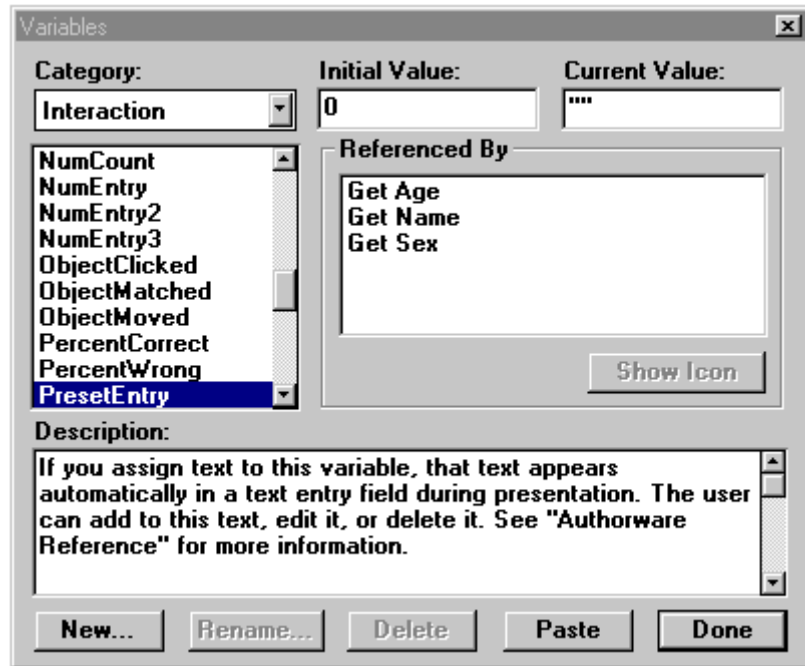


Another enhancement can be made at this level, and that is when the end-user re-enters a variable they have to enter the whole things from scratch at present. That is if I enter **spencer** as my name, and subsequently want to enter **spencer hudson**, I have to re-enter the whole thing. What I really want to do is just add my surname. This is a nice user-friendly touch, which can be done in Authorware. In essence we pre-load the **EntryText** variable with a value.

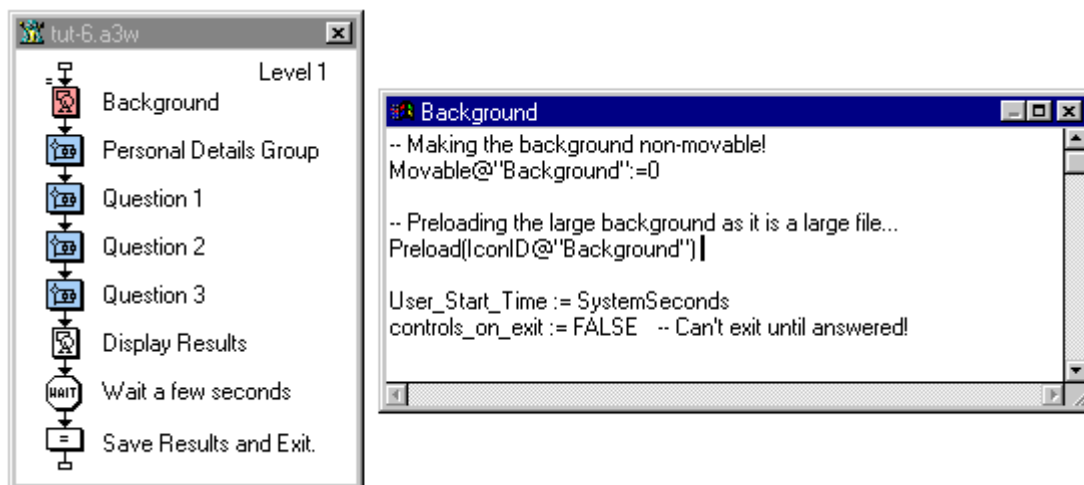
- Open each of the personal details groups in turn and add the following calculation
- Use **CTRL+=** to attach a calculation to the interaction icon



Remember, you need not remember the exact wording of a variable or a function. You can always access them through the menu and use the past command.



One other command is worth mentioning at this point, the function is called **Preload**. This function will preload any icon - this is particularly useful for large graphics. I have used the command to preload the opening graphic:

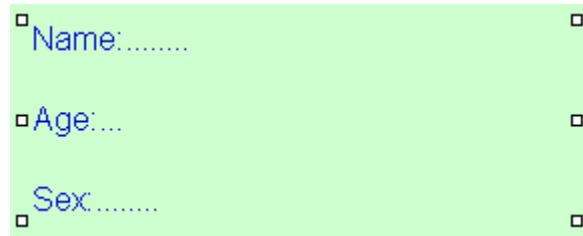


You will also notice that I have moved some of the icons and calculations around to make more efficient use of the available resources (i.e. icons). Efficient and effective use of available resources are key skill of any multimedia author!

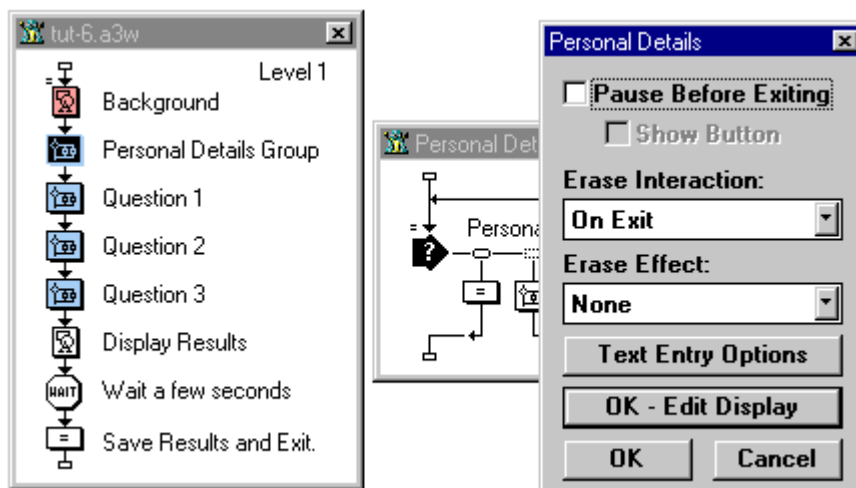
2) Defining tabs

Good alignment of elements is essential to any spatial orientated product (e.g. a building, paper-based). In tutorial five when getting the personal details from end-users tabs could have been used to arrange variables. This process will now be explained:

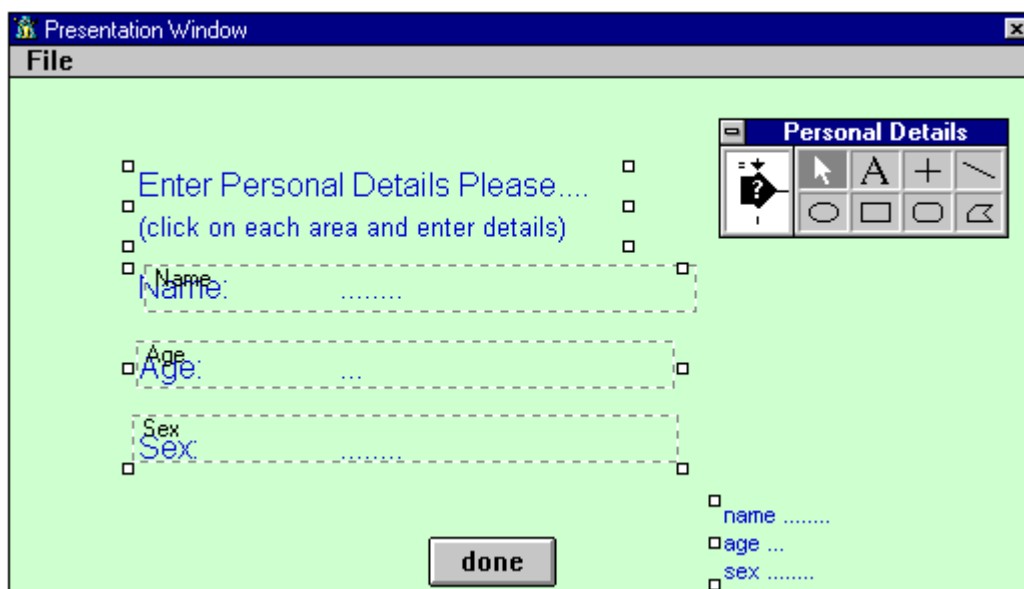
- Without spacing or tabs the personal details would look something like this:



- We will now define a tab and position the three variables (signified by the dots).
- First, open **Personal Details Group**, then double-click on the **Personal Details** interaction icon and finally select **OK - Edit Display**

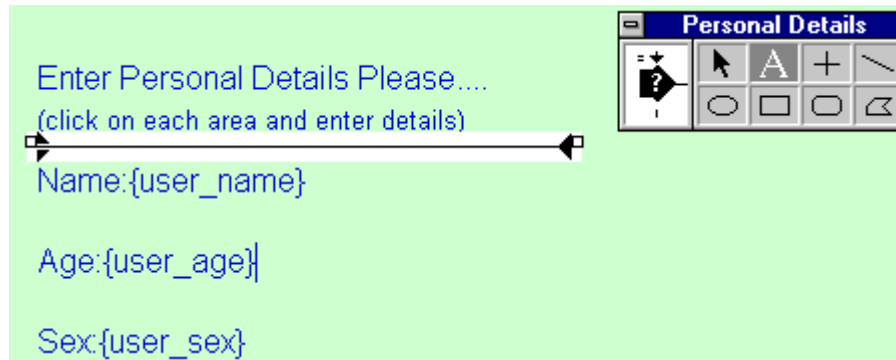


- You should now see something like this:

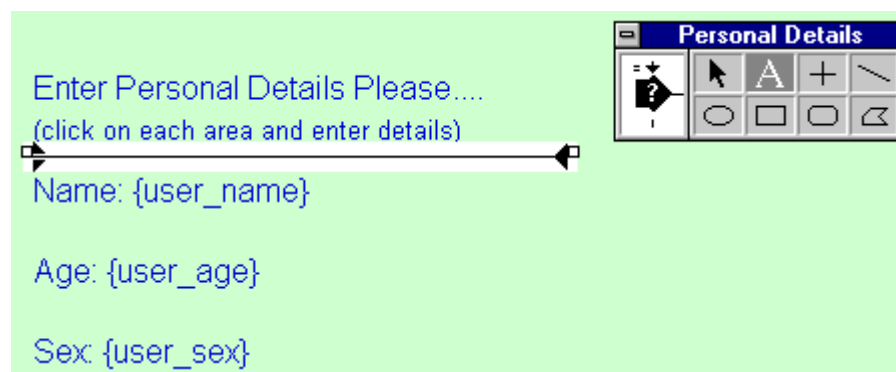


Note: For simplicity I will now remove the hot-spots

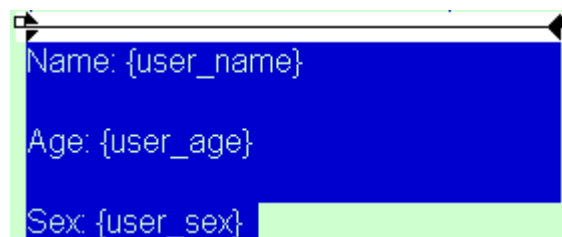
- Select the text tool (i.e. symbolised by the letter **A**) and click on the first hot-spot area
- You should now see something like this, no tabs define:



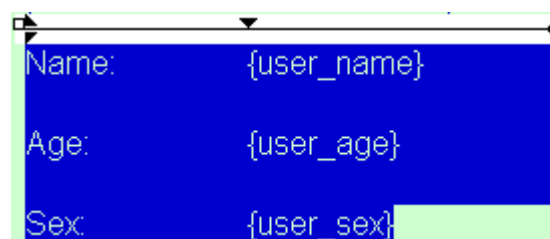
- You will notice a line white line with a black line running through it at the top of the text entry area. The black line is where you position the tabs. The are two types of tabs: (a) left align; and (b) right align
- Insert a tab between the :{ for each of the line. You should notice a little space inserted, we can now create tabs



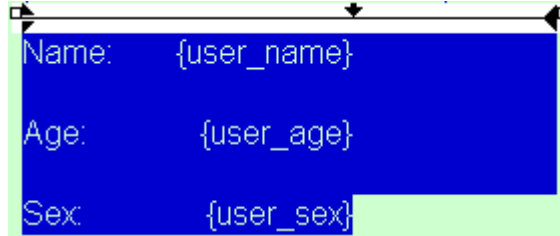
- First, select all of the text



- Now click on the black line, about half way along. You will notice all of the text lined up underneath where you have just added a tab



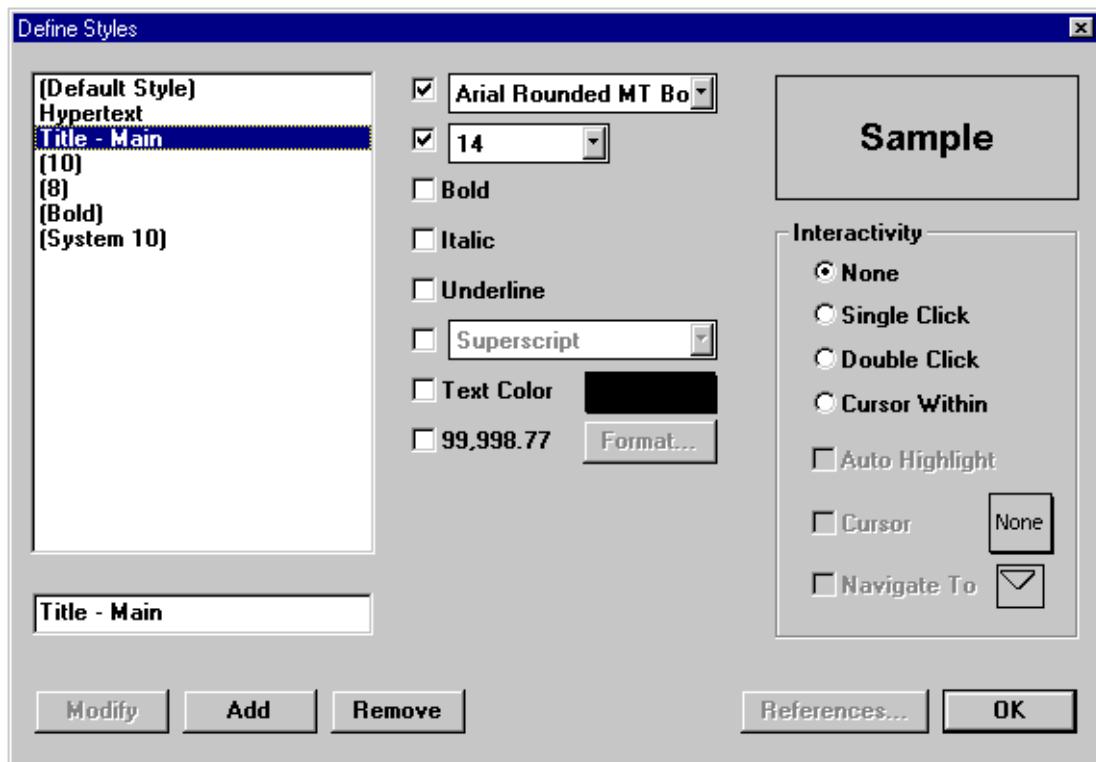
- While all of the lines you wish to effect are select you may click-on and move the tab to the left and to the right. To delete the tab move it off the end of the line (either to the left and off or the right)
- There is a second try of tab, right align. To create a right align tab simply click on the one we have already created. You will notice it change shape and the variable align up to the right



3) Defining fonts

One of the more underused features of Authorware is the facility for defining fonts. However, this can save a developer a great deal of time and effort. Indeed, we have already create one style of font in a previous tutorial, i.e. Hypertext. This is not the only use for defining fonts, we can use it to define whole sets of titles that are to be used in an application. The main advantage being that if we need to change style later on we can do this easily.

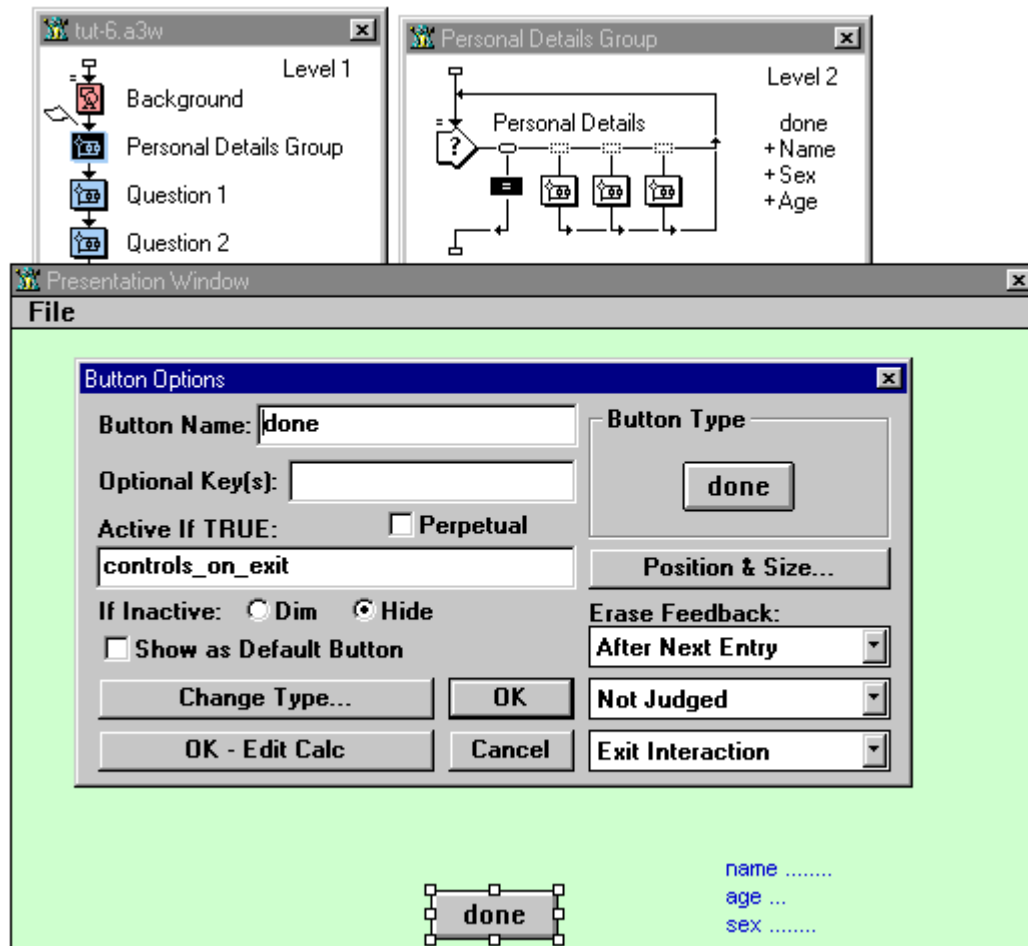
- Below, you can see that I have create a style **Title - Main** it is suggested that you create similar styles for your application



4) Defining buttons

One especially nice feature of Authorware is the facility to define and use your own buttons. This enables the designer to create very interesting interactions. In this example we will create a new default **done** button.

- Open the **Personal Details Group**, and double click on the **done** button response type

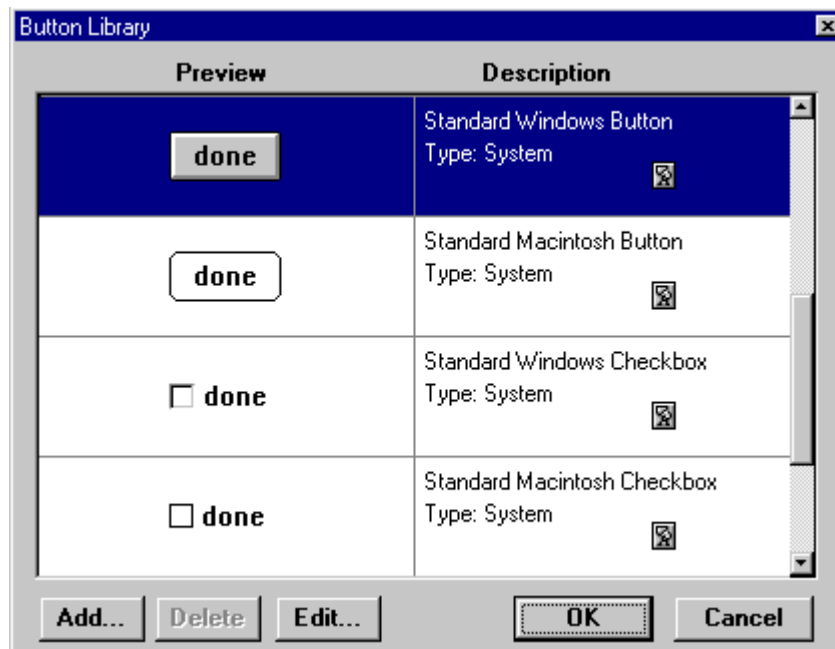


While we have the **Button Options** dialogue box open we will add a nice touch to the interaction. Rather than making the end-user have to click on the **done** button, we will allow them to press **enter**. To do this:

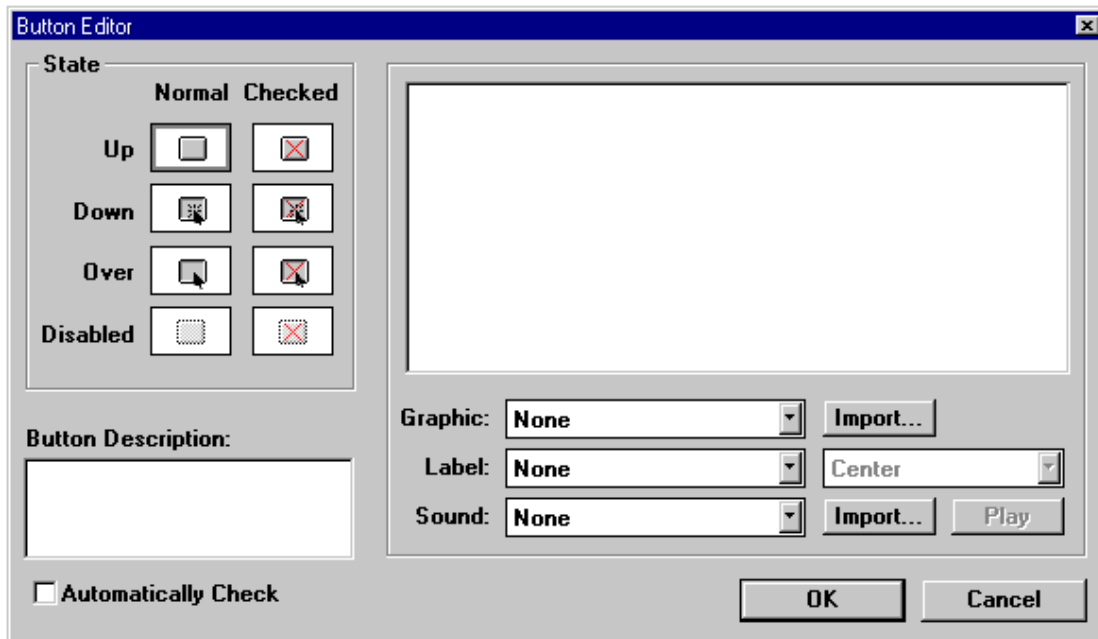
- In the box - shown above - labelled **Optional Key(s):** type **enter**



- Click on the **Button Type** it is on the top right, you should now see



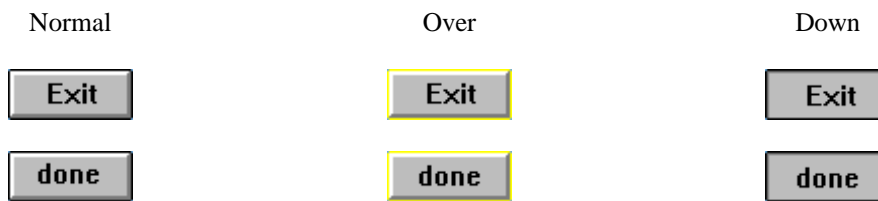
- Click on the **Add...** button
- You should now see the flowing dialogue box



Essentially in this dialogue box you can define different looks for a button depending upon its state. Each state can have a different graphic (i.e. change the look of the button), or a different sound attached to it.

For each state a different graphic, label and sound can be combined to create a button. There is also the facility (bottom right) to define a name for the button set.

Below is an example of buttons that I have created. The normal button is the same. The over button has a yellow border around it. Thus when the end-user's cursor goes over the button it gives the effect of being highlighted.



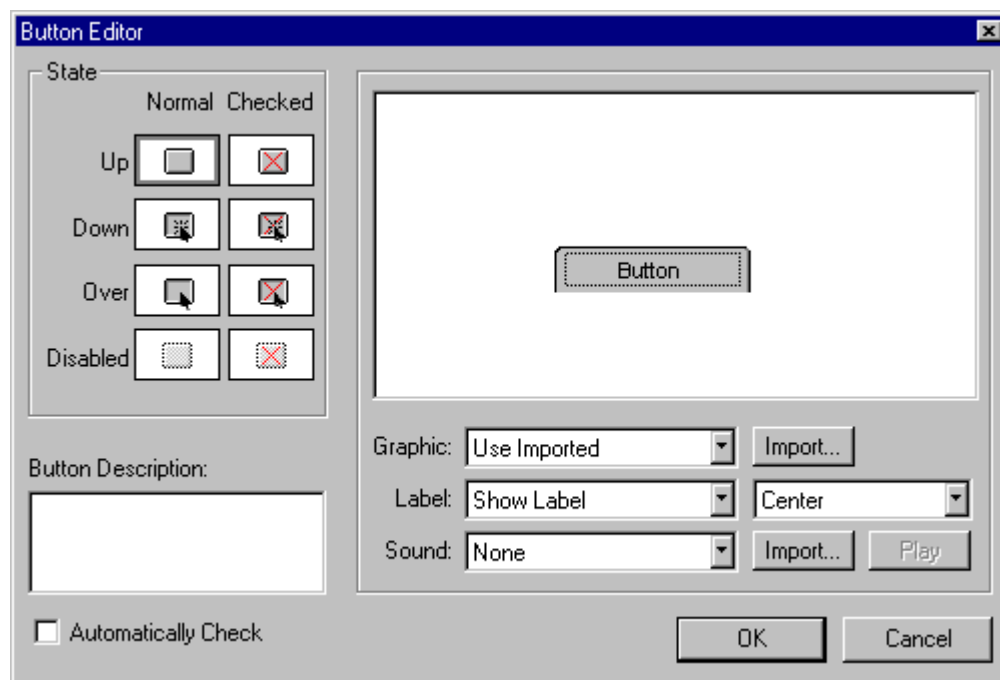
These button can be easily created in any graphics package. If you want to alter traditional buttons, one tip here is to use the Print Screen facility to capture the look of the button from Authorware (i.e. work thought the three different buttons, capture them and then edit them).

Here if my newly defined button, from the **Button Library**

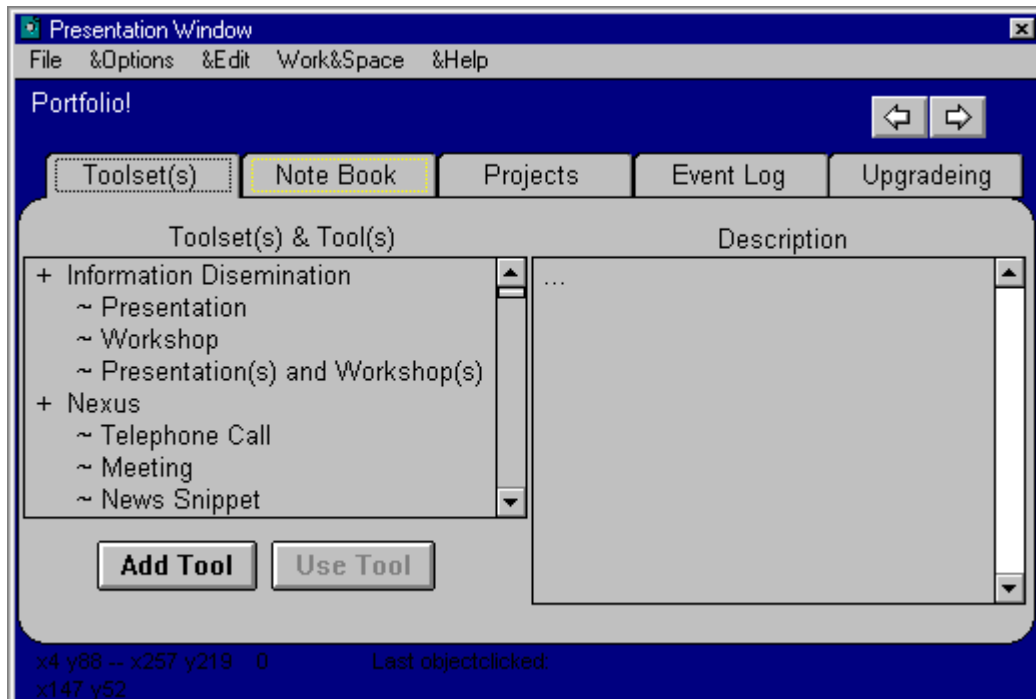


- Now redefine the other questions to use this button

Finally, when creating buttons you do not have to embed the label. That is you can create blank buttons, such as those available within Authorware. Thus when you define a button you leave the label off and let Authorware put it on for you. Below is an example, where I have created buttons that act as tabs. No labels have been put on the buttons, this is done by Authorware. Notice how in the Button Editor I have selected the option **Label: Show Label** together with **Centre** alignment (i.e. for the button title, in this example the word "Button").



Below is an example of how I used the 'tab' button that I defined. Notice that Authorware have added the different labels onto the buttons for me.



Summary

- CTRL+H = Align Objects
- CTRL+= = This is still a useful command.
- CTRL+X and CTRL+V = The standard windows 3.x Cut and Paste commands
- PrtScr = Capture whole screen (Print screen) and CTRL+V to paste it into a graphics package
- CTRL+PrtScr = Capture current window and CTRL+V to paste it into a graphics package

In your own time

Interaction Icon

- **Text Entry:** create a validation for the personal details entered, for example: age is a number which is between....; name is from a list of valid names, maybe read the names from a file.

Answer to last tutorials error

- The first error was in the **Save Results and Exit** calculation, it did not save the results from the third question. The second error was to do with the first question, the variable **controls_on_exit** was not set to false.