Lab project - Making Interactive Models

- Example 1. Make an interactive model that computes \( n^2 \), for \( n \) in the interval \([1, 4]\).

  ◎ Solution 1

  Manipulate\[n^2, \{n, 1, 4\}\]

  Once an interactive model is created, you can copy and run the model in a new notebook. You do not need to copy the code.

  We can indicate the domain of \( n \) in different ways (each way will generate a different type of control button, slider, popup menu, checkbox, etc.)

  Manipulate\[n^2, \{n, 1, 4, 0.25\}\]

  Manipulate\[n^2, \{n, 1, 4, 1\}\]

  Options for the controls can also be given within the specification for the variables.

  The option setting \texttt{ControlType \rightarrow type} attempts to use controls of the specified type. Possible control types include: Animator, Checkbox, CheckboxBar, ColorSetter, ColorSlider, InputField, Manipulator, PopupMenu, RadioButton or RadioButtonBar, Setter or SetterBar, Slider, Slider2D, TogglerBar, Trigger, and VerticalSlider. None. More information can be found in the Mathematica Documentation.

  The option setting \texttt{ControlPlacement \rightarrow pos} specifies that controls should be placed at position pos relative to expr. Possible settings for position are Bottom, Left, Right, and Top.

  Manipulate\[n^2, \{n, 1, 2, 3, 4, 5, 6, 7, 8\}, \texttt{ControlType \rightarrow SetterBar}\]

  Notice that the cell reevaluates itself continuously (the right cell bracket is constantly blinking), even when we do not change the position of the slider. You can confirm this by going to Evaluation\rightarrow Find Currently Evaluating Cell. This happens because the variable temp has its value changed during the evaluation (temp = temp\(^2\)), even if the value of n has not changed.

  ◎ Solution 2

  Manipulate[
    temp = n;
    temp = temp^2;
    temp, 
    \{n, 1, 4\}\]

  Notice that the cell reevaluates itself continuously (the right cell bracket is constantly blinking), even when we do not change the position of the slider. You can confirm this by going to Evaluation\rightarrow Find Currently Evaluating Cell. This happens because the variable temp has its value changed during the evaluation (temp = temp\(^2\)), even if the value of n has not changed.

  ◎ Solution 3

  The problem can be solved by making the global variable "temp" be local variable inside a Module. Nothing you do to local Module variables will cause reevaluating, because it is part of the definition of Module that values of local variables do not survive from one invocation to the next.
Manipulate[Module[{temp},
  temp = n;
  temp = temp^3;
  temp],
  {n, 1, 6}]

\textbf{Solution 4}

Manipulate[
  f[x_] := x^2;
  f[m],
  {m, 1, 4}]

\textbf{Solution 5}

Manipulate[Module[{g},
  g[x_] := x^2;
  g[n]],
  {n, 1, 4}]

The function \(g\) is now a local variable, so it does not cause any extra reevaluations. Notice that it is not defined outside of Module[...], so if you try to call \(g[3]\) say, somewhere below, the function \(g\) will not be recognized.

\(g[3]\)

\textbf{Solution 6}

Manipulate[
  f[x_] := x^2;
  f[n],
  {n, 1, 4}, TrackedSymbols \to \{n\}]

We can keep the function \(f\) global, without causing any reevaluations of \texttt{Manipulate}, if we explicitly indicate that the only variable whose values we should keep track of is the parameter \(n\) (by default, \texttt{Manipulate[...]} tracks both \(f\) and \(n\)). The example above reevaluates only when \(n\) changes its value as a result of moving the slider.

\textbf{Solution 7}

\(h[x_] := x^2\)

Manipulate[
  \(h[x]\),
  \(x, \{1, 4\}\)]

We can also define the function \(h(x) = x^2\) globally, before \texttt{Manipulate[...]. At first glance, it looks like everything works well and without causing any reevaluations of \texttt{Manipulate}.

The downfall is that the definition of the function \(h\) (which is called later in the body of \texttt{Manipulate}) is not saved together with the output of \texttt{Manipulate}. To see this, open a new Mathematica notebook, and set ‘Evaluation->Notebook’s Default Context’ to ‘Unique to this notebook’. Then copy ONLY the output of \texttt{Manipulate} (that is, ONLY the interactive model, absolutely NO code) in the new notebook. Try changing the slider. What do you notice?

\textbf{Solution 8}

\(h[x_] := x^2\)

Manipulate[
  \(h[n], \{n, 2, 4\}, \text{SaveDefinitions \to True}\)]

The option ‘SaveDefinitions \to True’ forces any function definitions used by \texttt{Manipulate} to be saved with the output. The output can be copied and directly run in a new notebook. Try it!

- **Exercise 1.** Make an interactive model that reads the value of two variables, total and sum, represented by two different controls, and outputs their sum, total + sum. Does the example below do that?

\texttt{Manipulate[}
  \(\text{total = total + step;}
  \{\text{step, total}, \{\text{total, 0}, -1000, 1000, 1\}, \{\text{step, 0}, -10, 10, 1\},
  \text{FrameLabel \to \"Good or bad\"}]\]

- **Exercise 2.** A way to compute the sum of the first \(m\) positive integers in Mathematica is to do a \texttt{For} loop, like the one written below. Turn the code below into an interactive model where \(m\) will be allowed to take any values in the set \{10, 20, 30, 40, 50\}.

\(m = 5;\)

\(s = 0;\)

\texttt{For[\(i = 1, i \leq m, i++\), \(s = s + i\); \s]}

- **Exercise 3.** Interactive models with graphs. This is the exercise from last time, plus some additional tasks.

Open the Mathematica notebook MetroStationMap. The variable called \texttt{Metro} is a list of 5 graphs with the metro networks in Seoul, New York, Paris, Mexico City and London. The vertices of the graphs are labeled with the names of the stations.

\texttt{NameCity = \{\"Seoul\", \"New York\", \"Paris\", \"Mexico City\", \"London\"\]}

Notes7.nb | 3

Notes7.nb | 4
(a). Write an interactive program using Manipulate[...] that has a control button with five values 1, 2, 3, 4, 5. When the user selects a given value, you display the name of the city and the corresponding metro map. Because your program uses some global variables defined outside Manipulate (like Metro), you should also set SaveDefinitions->True.

(b). Expand the model from part (a) with a second control which gets populated with the vertices of the corresponding graph. When the user selects a vertex from the list, you show the corresponding metro station highlighted in green (or with increased size) on the metro network. Make the control type a PopupMenu.

(c). Make an interactive model from part (b) with nodes v and u as parameters, that highlights the shortest path from v to u, if there exists such a path in the graph, otherwise it only highlights the two vertices. Use the function FindShortestPath[...].

(d). Make an interactive model that highlights a vertex v together with its parents and children (i.e. if there is an edge v→u, then v is called a parent node and u a child node). The function NeighborhoodGraph[...] might be of help.

(e). A Metro hub is a metro station which has direct connections to five or more other metro stations. Make an interactive model where the metro hubs are highlighted with VertexSize->1.2 Use for example VertexInDegree[...] to count the number of incoming edges for each node. You may also need to use the function Sort[...] (check also the Mathematica documentation on Sort).

(f). Optional: Combined Interactive Model: Parameters: nodes v and u, and an extra parameter "option" with values {b,c,d,e}. If "option" b is selected, then the model outputs the same thing as part (b), if "option" c is selected, then the model outputs the same thing as part (c) etc. If you skipped some part, say (d), then you should allow only two values for "option", {b,c,e}. You may need to use some conditional statements like If or Switch.