# IFB104 GUI Workshop Exercise: Batmobile Ignition

In this exercise you will use the tkinter module to develop a Graphical User Interface that uses radio buttons in two different groups to control the text displayed in a label.

Motivation: One of the many memorable features of the 1966 Batman television series was the elaborate ignition sequence needed to start the nuclear-powered Batmobile, which always proceeded as follows:
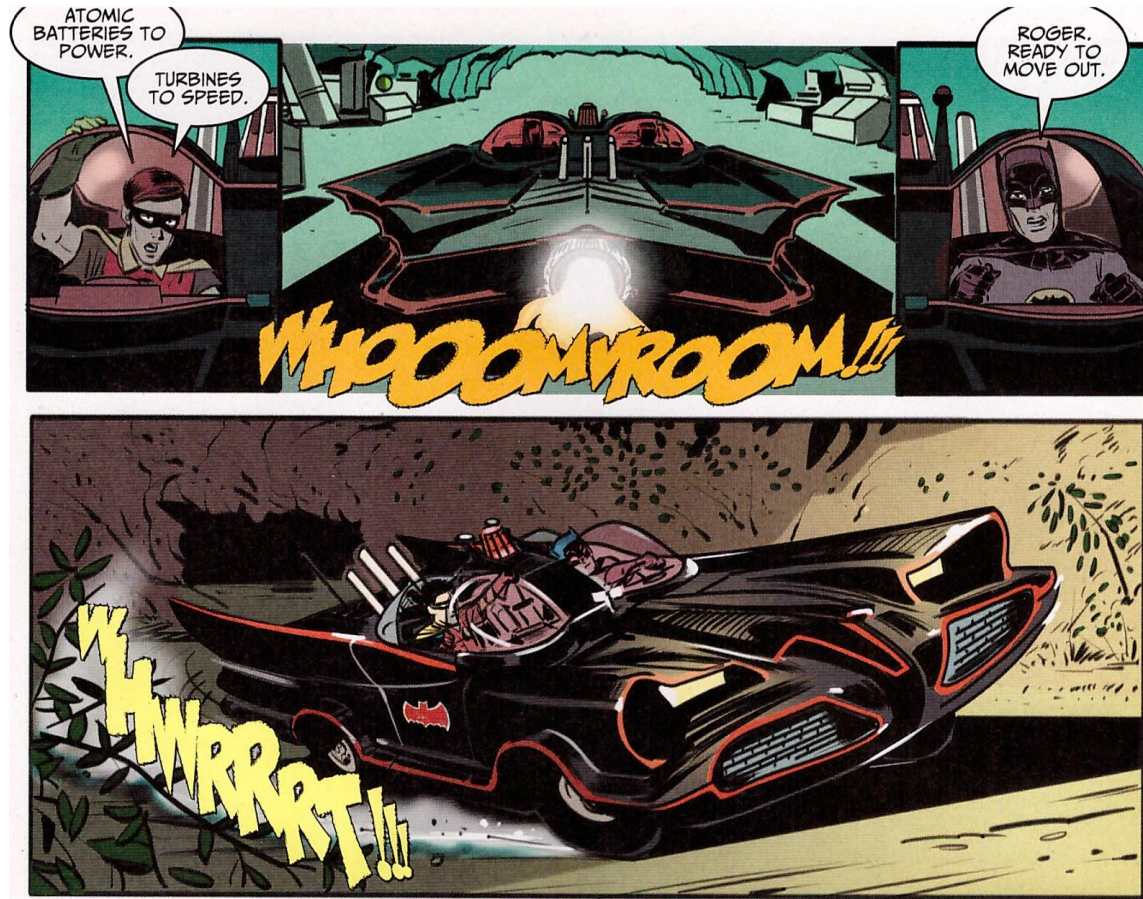

Having strapped themselves in, Robin reads out the status of the Batmobile's atomic batteries and turbines, after which Batman starts the ignition and with a roar of flame the Batmobile begins its high-speed journey from the secret Batcave under stately Wayne Manor to Gotham City.

In this exercise you will duplicate part of the Batmobile's dashboard display using label and radio button widgets. When complete, your program should create a GUI which initially looks as follows.
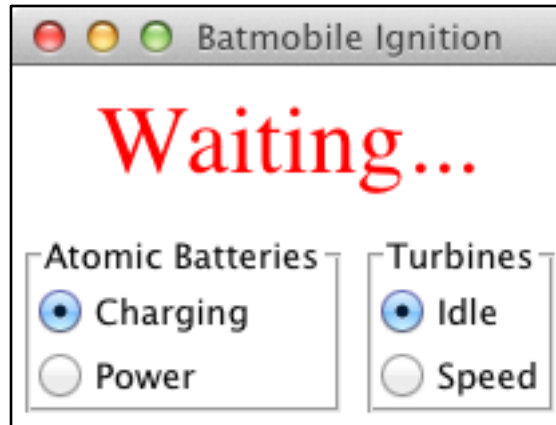
This window has a large label at the top, and four radio buttons at the bottom. In addition, we have put the radio buttons inside two "label frame" widgets, but you may choose to omit this nicety at first.

The radio buttons reflect the state of the Batmobile's atomic batteries and turbines. In the initial state the ignition system is in a "waiting" mode. Changing just one of the pairs of radio buttons has no effect on the Batmobile's status:
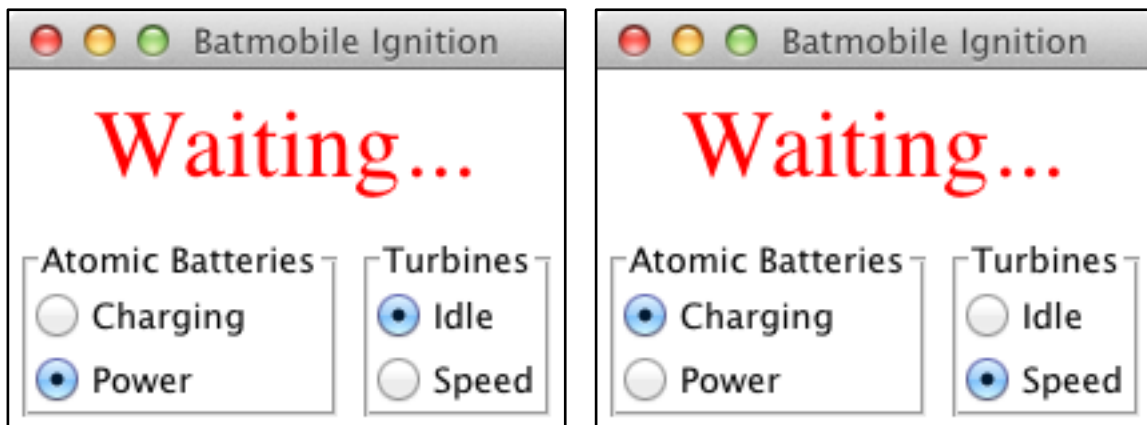
Only when the atomic batteries are set to 'Power' **and** the turbines are set to 'Speed' is the Batmobile 'ready to move out':

To complete this exercise you'll need to:

- Use the `tk` and `Label` constructors to create the root window and the label up the top, whose text is initially 'Waiting...'.

- Define two `BooleanVar` variables to represent the state of the two pairs of radio buttons.

- Use the `Radiobuttons` constructor to create the four radio buttons. Each pair of buttons should have its `variable` parameter linked to one of the Boolean variables. The `value` parameters associated with the buttons should tell us whether or not the Batmobile's atomic batteries and turbines are set appropriately for ignition.

- Optionally put the radio buttons into separate `LabelFrame` widgets to create the neat layout shown above.

- Define a function which is called whenever one of the radio buttons is pressed. This function should 'get' the values of the Boolean variables (which are set by the radio buttons) and use them to update the label widget's `text` parameter to either 'Waiting...' or 'Ready!' as appropriate. (In our example we have also changed the foreground colour attribute, `fg`.)