PIC 16: Final (due 6/11 at 8am California time)

How should your answers be submitted?

- Download `images.py` and `PyQt.py`.
- Replace `mjandr` by your name.
- Answer each of the exam questions in the corresponding file.
- Submit these two files to CCLE *before* the deadline.

  I advise you to try and submit your work at least an hour before the deadline in order to avoid internet connection issues.

1 Images (12 points)

Download `barrel.jpg` and `jb.jpg`; do *not* rename these files. Type your solution code in `images.py`.

So that you don't spend a long time trying to find various numbers... the only integer literals I used in my solution were `-1`, `2`, `150`, `165`, and `400`.

(a) Have your code produce a file called `jb_barrel_A.jpg`.

The image of James Bond should be placed in the **center** of the image of the barrel.

(b) Have your code produce a file called `jb_barrel_B.jpg`.

Shift James Bond further down in this image, and chop off his legs so that he fits.

(c) Have your code produce a file called `jb_barrel_C.jpg`.

Use masking to fit James Bond elegantly inside the barrel.

2 PyQt (12 points)

I have provided you with `PyQt.py`. Running it produces a window entitled "My Final Widget". Your job is to add to this `QWidget` so that:

- it always has a white background regardless of the default for the operating system;
- clicking the mouse specifies vertices of a polygon;
- blue edges are drawn as more vertices are specified;
- double-clicking "closes up" the polygon;
- the click **after** a double-click will erase the previous polygon **and** start a new one.

See `PyQt_demo.mov` for how the file should run when you finish.
(Remember that the black circles are created by QuickTime player to show you my mouse clicks.)

Here is some advice on how to proceed.

- For the white background, draw a white rectangle of the correct shape (as on the homework).
- Store the vertices of the polygon in a list of tuples. You may want to use an instance variable `self.vs`. You should *not* need to store the edges; they can be deduced from the list of vertices.
- To draw `n` edges, use `https://doc.qt.io/qt-5/qpainter.html#drawLine-2` `n` times (copy and paste the address if the link does something weird to the #). This is the least confusing function available to you: it takes in four integers.
- Suppose that between a double-click and the next click you resize the window. The polygon should *not* disappear as a consequence of this resizing. In order to handle this correctly I introduced a boolean instance variable `self.doubleClicked`.