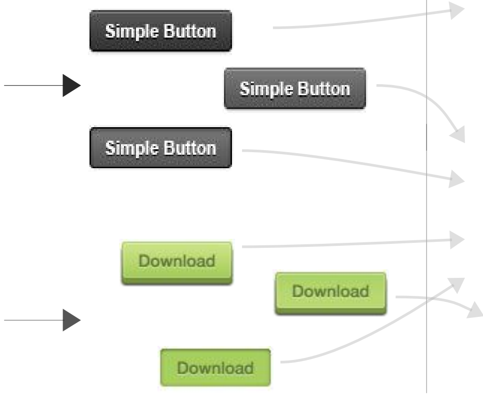
summer buttons

Push & Toggle Button

Psds (we need high quality buttons @3 states)

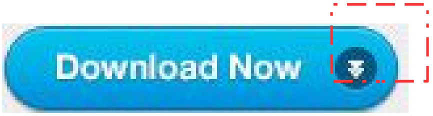
Extract **jpg**'s from photoshop (minimum margins if possible)



```
pb = uicontrol('Style','pushbutton', 'Units','pixels',
'Position',pos,'Callback',@(a,b)disp('!!'),
'CData', img);


findjobj(pb) %expose java object via external utility (included)
newBorder = javax.swing.border.LineBorder(lineColor,thickness,roundedCorners);
%adjust border and focus or remove completely

pbj.setRolloverIcon(javax.swing.ImageIcon(im2java(img2)));
pbj.setPressedIcon(javax.swing.ImageIcon(im2java(img3))); %use appropriate
java callbacks for button's states

%for toggle buttons you need the default matlab callback (for the two states)
%e.g. toggleCallback(hobj,~)

%and another java callback for mouse rollover
pbjt.setRolloverIcon(javax.swing.ImageIcon(im2java(img3)));
```

**Default setting for exported layer is a transparent checkboard (white). In case figure/axes is colored you need to adjust accordingly

Arbitrary shaped Buttons + background

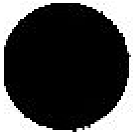


Background is neither white nor uniformly gray. Adjust the exported image (paint on mask) or change color in gui's axe/figure (not always applicable -what if we want our gui to be pink?)

A cool & round power button needs work ...

First we need to find the appropriate layer for changing the color to match with the axes/figure/image we want to create for the button (in this case its a matlab logo with irregular shading. (All the psds I found online have plenty of layers and it's easy(ish) to find a color layer)

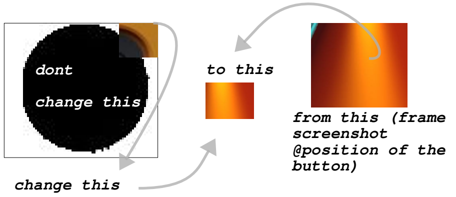
Select a Darker hue for the “pushed” state



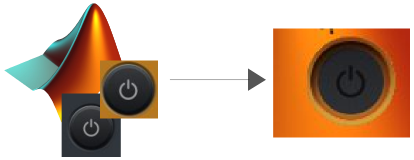
Finally, we need a mask for the button (I chose the outer layer, paint it black and exported it)

**If you dont change the color layer, you can still use the same technique but with less (esthetically) pleasing results, due to different colors within the edges of the button.

```
pb = uicontrol('Style','togglebutton', 'Units','pixels',
'Position',pos,'Callback',@(a,b)disp('!!'));

%use the mask we created in photoshop for marking the button arrea
mask=rgb2gray(mask);
mask=mask>20; %not 0 due to mask color aliazing

% take a screenshot of the parent figure (you can use it for axes too) to
know the colors underneath the button
F = getframe(ax,pos);

for idx = 1 : 3
   rgb = img11(:,:,idx);
   pImage = double(F.cdata(:,:,idx));  % extract part of the image
   rgb(mask) = pImage(mask);    % set the white portion of the image to the parent
   img11(:,:,idx) = rgb;        % substitute the update values
...
... (do the same for all three images)
end
```



from this (frame screenshot @position of the button)



```
set(pb, 'CData', img11); %put the new icon on cdata of the button
pb = findjobj(pbtt);pbj.Border = [];pb.FocusPainted=0; %remove border &focus

%we need the standard matlab toggle callback for changing images when the
button is clicked e.g.toggleCallback(hobj,~)

%And we need to set an intermediate button state to complete the optical
illussion, which is %available only via pure java callback
set(pbj,'MousePressedCallback',@istate);
function istate(hobj,~);set(pbtt, 'CData', img31);end
```