

Hands-on Activity: Handling Stack Geometry and Device Orientation

-Demonstrating techniques for resizing and repositioning objects when the screen size changes.

1. Create a New Stack. Don't bother resizing.
2. Draw a rectangle graphic, name "outerFrame", give it thick border and a bright line color.

Card Script:

```
on updateCardLayout
  set the width of graphic "outerFrame" to the width of this stack
  set the height of graphic "outerFrame" to the height of this stack
  set the topLeft of graphic "outerFrame" to 0,0
end updateCardLayout
```

Try out this script in the message box.

But we want this to happen automatically when the screen is resized. For this use the `resizeStack` message. The `resizeStack` message receives arguments when the message is sent: the new width of the stack, new height of the stack, old width, old height. We're only interested in the new dimensions. So modify the stack script like this:

```
on resizeStack pNewW,pNewH
  updateCardLayout pNewW,pNewH
  # pass width and height as args to save some typing in the called handler
end resizeStack
```

```
on updateCardLayout pWidth,pHeight
  set the width of graphic "outerFrame" to pWidth
  set the height of graphic "outerFrame" to pHeight
  set the topLeft of graphic "outerFrame" to 0,0
end updateCardLayout
```

Now apply and try resizing the stack. The frame graphic is reset to the desired dimensions with each resize.

3. Add a field in the center of the stack. Name "screenInfo".

We want this field to always be $\frac{2}{3}$ the width and half the height of the stack, and always positioned in the center of the card.

```
on updateCardLayout pWidth,pHeight
  # set outer frame
  set the width of graphic "outerFrame" to pWidth
  set the height of graphic "outerFrame" to pHeight
  set the topLeft of graphic "outerFrame" to 0,0

  # set size and position of screenInfo field
  set the width of fld "screenInfo" to 2 * pWidth / 3
  set the height of fld "screenInfo" to pHeight / 2
  set the loc of fld "screenInfo" to loc of this cd
end updateCardLayout
```

4. Now report information about the screen size:

```
on updateCardLayout pWidth,pHeight
  # set outer frame
  set the width of graphic "outerFrame" to pWidth
  set the height of graphic "outerFrame" to pHeight
  set the topLeft of graphic "outerFrame" to 0,0

  # set size and position of screenInfo field
  set the width of fld "screenInfo" to 2 * pWidth / 3
  set the height of fld "screenInfo" to pHeight / 2
  set the loc of fld "screenInfo" to loc of this cd

  # report size information
  put "the screenRect:" && the screenRect into fld "screenInfo"
  put return & "the working screenRect:" && the working screenRect after fld "screenInfo"
  put return & "the effective working screenRect:" && the effective working screenRect after fld
"screenInfo"
end updateCardLayout
```

5. Let's see how this looks on the mobile simulator or device.

Stack script:

```
on openStack
  if the environment is "mobile" then
    mobileSetAllowedOrientations "portrait,portrait upside down,landscape left,landscape right"
  end if
end openStack
```

Set up for deploying on iOS and Android in the Standalone Application Settings. (Remember to make the application ID unique. Just adding dot (.) plus the stack name afterward should be the trick.

Try it on the mobile device or simulator, including rotating the device. Note how (if!) the numbers change.

6. Show some other information in the field.

```
  Add to updateCardLayout:
  if mob() then
    put "current device orientation:" && mobileDeviceOrientation() \
      into line 4 of fld "screenInfo"
    put "current interface orientation:" && mobileOrientation() \
      into line 5 of fld "screenInfo"
    put "pixel density of the screen:" && mobilePixelDensity() \
      into line 6 of fld "screenInfo"
  end if
```

7. Lock screen rotation

- Add switch button widget and label field with text "lock rotation" next to it; font size 18; position directly above field.

Script:

```
on hiliteChanged
  if mob() then
    if the hilite of me then
      mobileLockOrientation
    else
      mobileUnlockOrientation
    end if
  end if
end hiliteChanged
```

8. Now we have to reposition the checkbox when we update the screen

Add to updateCardLayout:

```
# set position of switch widget
set the bottom of widget "lockOrientation" to the top of fld "screenInfo" - 4
set the left of widget "lockOrientation" to the left of fld "screenInfo"
set the left of fld "lockLbl" to the right of widget "lockOrientation" + 8
set the bottom of fld "lockLbl" to the bottom of widget "lockOrientation"
```

9. You can see that your app can still be aware of device orientation, even if the orientation of the app interface is locked. Try adding this to the card script:

on orientationChanged

NOTE: this message is sent BEFORE automatic interface rotation takes place

```
if mob() then
  put "current device orientation:" && mobileDeviceOrientation() \
    into line 4 of fld "screenInfo"
  put "current interface orientation:" && mobileOrientation() \
    into line 5 of fld "screenInfo"
  put "pixel density of the screen:" && mobilePixelDensity() \
    into line 6 of fld "screenInfo"
end if
end orientationChanged
```