

Extending touchgui: adding a worldspawn button

```
def worldspawn (name, tap):
    pygame.display.update ()
    if tap == 1:
        print ("worldspawn called", name, tap)

def glyphs ():
    return [touchgui.text_tile (dark_grey, light_grey, white, mid_g
        's', touchgui.unityY (0.05),
        touchgui.posX (0.5), touchgui.posY
        100, 100, worldspawn, "worldspawn")
```

Extending touchgui: adding a worldspawn button

```
def main ():
    global players, grid, cell_size

    pygame.init ()
    if full_screen:
        gameDisplay = pygame.display.set_mode ((display_width, disp
            FULLSCREEN)
    else:
        gameDisplay = pygame.display.set_mode ((display_width, disp

    touchgui.set_display (gameDisplay, display_width, display_heigh
    controls = buttons () + glyphs ()

    gameDisplay.fill (touchguipalate.black)
    while True:
        grid = button_grid (cell_size)
        forms = controls + grid
        touchgui.select (forms, event_test, finished)
```

Extending touchgui: adding a worldspawn button

- at this point the call back worldspawn is in place
 - worldspawn can be made turn on worldspawn mode
- then callback can be modified to detect this mode and add the appropriate tile
- hint it might be sensible to use an indirect function
 - empty_cell_click
 - which is initially set to empty_to_wall
 - and can be changed to empty_to_worldspawn
- this technique allows touchmap to be extended to place hellknights, imps, ticks and pickups

Local images

- local images should be placed inside the touchmap directory
 - for example: touchmap-0.1/images
 - these images are kept in the source directory touchmap-0.1
- our build occurs in: build-touchmap
 - therefore the Makefile.am needs to have rules inside it to copy the images from the into the build directory

Build directory

- should be treated as volatile
 - can be destroyed and created
- therefore all assets and source files **must** be kept in the `touchmap-0.1` directory

autotools

- are used to configure and maintain the building rules
 - see `Makefile.am` and `configure.ac`
- the file `Makefile.am` contains the building rules
- in our case we just need extend `Makefile.am` to copy the image from the source directory into the build directory

autotools

- `touchmap-0.1/Makefile.am`

```
all: doorh.png doorv.png doorh-bw.png doorv-bw.png hinge.h
    wallh.png wallv.png wallh-bw.png wallv-bw.png \
    newname.png

newimage.png: $(srcdir)/images/newimage.png
    °convert -resize 100x100 $< $@
```

autotools

- `touchmap-0.1/Makefile.am`

```
all: doorh.png doorv.png doorh-bw.png doorv-bw.png hinge.h
    wallh.png wallv.png wallh-bw.png wallv-bw.png \
    newname.png

newimage.png: $(srcdir)/images/newimage.png
    cp -p $< $@ # this line must start with a tab character
```

- notice that ° needs to be replaced by a single tab character
 - you might need to alter preferences in gedit to allow you to add a tab character