

```
1: # Fragment/split an image into tiles (initially, 512x512)
2: # Brygg Ullmer, Clemson University
3: # Begun 2023-03-22
4:
5: import sys
6: sys.path.append('c:/git/pgzero-xi')
7:
8: from enoContentResolver import *
9: from enoTiledImg import *
10: from enoTiledImgNav import *
11:
12: WIDTH, HEIGHT=1920, 1080
13: eyu = 'https://enodia.computing.clemson.edu/'
14: #ecr = enoContentResolver(eyu)
15: #addrL = ecResolv.getFirstContentMatchLocal('SC')
16:
17: eti = enoTiledImg()
18: etinav = enoTiledImgNav(eti)
19: #eti.imgPos = (-10000, 0)
20: multiresLevel = 2
21:
22: eti.adjustWindowPlacement(WIDTH, HEIGHT)
23:
24: tmdn = 'resources/ecce001/loc/maps/sc/1970'
25: eti.loadTmap(tmdn, multiresLevel)
26: etinav.constructIndexGui()
27:
28: def draw():
29:     global eti, cl
30:     screen.clear()
31:     eti.draw(screen)
32:     etinav.draw(screen)
33:
34: def on_key_down(key): etiav.on_key_down(key)
35: def on_key_up(key): etiav.on_key_up(key)
36: def on_mouse_down(pos): etiav.on_mouse_down(pos)
37: def on_mouse_move(rel, pos): etiav.on_mouse_move(rel, pos)
38: def on_mouse_up(): etiav.on_mouse_up()
39: def update(): etiav.update()
40:
41: def on_tag_enters(tagId): etiav.on_tag_enters(tagId)
42: def on_tag_exits(tagId): etiav.on_tag_exits(tagId)
43: def on_magn_moves(magnVal): etiav.on_magn_moves(magnVal)
44: def on_cap_touch(capTouchId): etiav.on_cap_touch(capTouchId)
45: def on_cap_release(capTouchId): etiav.on_cap_release(capTouchId)
```

```
23: class enoTiledImg:
24:     PIL.Image.MAX_IMAGE_PIXELS = 21000 11000
25:     tileDim = (512, 512)
26:     animTween = 'accel_decel'
27:
28: def animBottom(self):
29:     self.animPrefatory()
30:     isy = self.getImageSize()[1]
31:     x, y = self.imgPos;
32:     y = self.screenDim[1] - isy
33:     self.lastImgPos = self.imgPos
34:     self.imgPos = (x, y)
35:     animate(self.imgActor, pos=self.imgPos, duration=self.animDuration,
36:             tween=self.animTween, on_finished=self.animationFinishedCB)
37:
38: self.animationActive = True
```

```
18: class enoTiledImgNav:
21:     shiftPressed    = None
22:     cursorsPressed  = None
23:     mouseDown       = None
25:     bignudge        = 768
26:     smallnudge       = 50

80:     def __init__(self, eti):
81:         self.eti      = eti
82:         self.shiftPressed = False
83:         self.mouseDown = False

244:     def on_mouse_down(self, pos):
245:         self.mouseDown = True #pan
250:         for actorName in self.indexFnActors:
251:             a = self.indexFnActors[actorName]
252:             if a.collidepoint(pos):
253:                 print("touch detected:", actorName)

208:     def on_key_down(self, key):
209:
210:         if key == keys.LSHIFT or key == keys.RSHIFT:
211:             self.shiftPressed = True
212:
213:         if self.shiftPressed:
214:             nudge = self.smallnudge
215:             if key is keys.LEFT: self.eti.shiftImg(-nudge, 0); self.cursorsPressed['L'] = True
216:             if key is keys.RIGHT: self.eti.shiftImg( nudge, 0); self.cursorsPressed['R'] = True
217:             if key is keys.UP:    self.eti.shiftImg(0, -nudge); self.cursorsPressed['U'] = True
218:             if key is keys.DOWN: self.eti.shiftImg(0,  nudge); self.cursorsPressed['D'] = True
219:         else:
220:             nudge = self.bignudge
221:             if key is keys.LEFT: self.eti.animImg(-nudge, 0); self.cursorsPressed['L'] = True
222:             if key is keys.RIGHT: self.eti.animImg( nudge, 0); self.cursorsPressed['R'] = True
223:             if key is keys.UP:    self.eti.animImg(0, -nudge); self.cursorsPressed['U'] = True
224:             if key is keys.DOWN: self.eti.animImg(0,  nudge); self.cursorsPressed['D'] = True
226:             if key == keys.T: self.eti.animTop()
227:             if key == keys.B: self.eti.animBottom()
228:             if key == keys.L: self.eti.animLeft()
229:             if key == keys.R: self.eti.animRight()

233:     def on_key_up(self, key):
234:         if key == keys.LSHIFT or key == keys.RSHIFT:
235:             self.shiftPressed = False

275:     def update(self):
276:         nudge = self.smallnudge
277:
278:         if self.eti.animationRunning: self.eti.animUpdateImg() #if animation underway
279:
280:         if self.isCursorPressed() and self.shiftPressed:
281:             if self.cursorsPressed['L']: self.eti.shiftImg(-nudge, 0)
282:             if self.cursorsPressed['R']: self.eti.shiftImg( nudge, 0)
283:             if self.cursorsPressed['U']: self.eti.shiftImg(0, -nudge)
284:             if self.cursorsPressed['D']: self.eti.shiftImg(0,  nudge)
```