

PyGame

- PyQuest demo

Modules I used

- Display
- Event
- Font
- Surface, Image
- Key, Mouse
- Mixer, Music
- Rect
- Sprite
- Time

Modules I didn't use

- **Cdrom**
- **Cursors**
- Draw
- **Joystick**
- **Movie**
- **Overlay**
- **Sndarray**
- **Surfarray**
- Transform

PyGame time savers

- Event queue
- Groups
- Sound

“Classic” architecture

```
while True:  
    handle user input  
    update sprite locations  
    handle interaction (collisions)  
    update display
```

User input

```
for event in pygame.event.get(pygame.KEYUP):  
    if event.key == pygame.K_ESCAPE:  
        sys.exit(0)
```

```
for event in pygame.event.get(pygame.MOUSEBUTTONDOWN):  
    if event.button == 1:  
        self.shots.add(Shot(...))
```

```
pygame.event.clear()
```

Update sprite locations

- “the monkey tutorial”

```
group.clear()  
group.update()  
group.draw()
```

(how it really looks)

```
class MultiGroup:
    def __init__(self, *args):
        self.L = list(args)
    def __getattr__(self, attr):
        def f(*args, **kwargs):
            for group in self.L:
                getattr(group, attr)(*args, **kwargs)
        return f
```

```
clock.tick(60)
group = MultiGroup(group1, group2, ...)
group.clear()
group.update()
group.draw()
```

Collisions

```
sprite.groupcollide(group1, group2, kill1, kill2)
```

```
sprite.spritecollide(sprite1, group1, kill)
```

Rect methods

```
clamp  
move  
collidirect [, point, list, dict]  
clip  
union  
contains
```

Final touches

- Text, sound

```
def render_bg(s):  
    text = font.Font(None, 36).render(s, True, (10, 10, 200))  
    textpos = text.get_rect(centerx=background.get_width()/2)  
    background.blit(text, textpos)
```

```
mixer.music.load('arg.xml')  
mixer.music.play(-1)  
def play_sound(filename, volume=0.5):  
    # you'd want to cache these in a real app...  
    snd = mixer.Sound(filename)  
    chan = mixer.find_channel(True)  
    chan.set_volume(volume)  
    chan.play(snd)  
    return chan
```

Some tips

- Use a separate class for explosions or other “different states” of your object
- Use event queue, not “instantaneous” key, mouse methods
 - But use it right: grab a copy of all events with `event.get`; don't use `event.clear`, or you could clear events you never had a chance to process
 - (yeah, sorry, my sample code is a Bad Example here)

I wish Pygame had

- Vector class
 - Math vector, not java vector
 - <http://www.pygame.org/pcr/vector/index.php>
 - sucks

Also of interest

- PyGame for non-games
 - http://media.libsyn.com/media/awaretek/Python411_060317_PyGame.mp3
- Sprite tutorial
 - <http://www.pygame.org/docs/tut/SpriteIntro.html>
- Solar Wolf
 - <http://www.pygame.org/shredwheat/solarwolf/>
- PyGame Code Repository
 - <http://www.pygame.org/pcr/>
- How To Build a Game In A Week From Scratch With No Budget
 - <http://www.gamedev.net/reference/articles/article2259.asp>
- Xquest
 - <http://www.swallowtail.org/xquest/>
- 3D engines
 - Panda3D, PyOgre