http://www.pygame.org/

PYGAME

Módulos

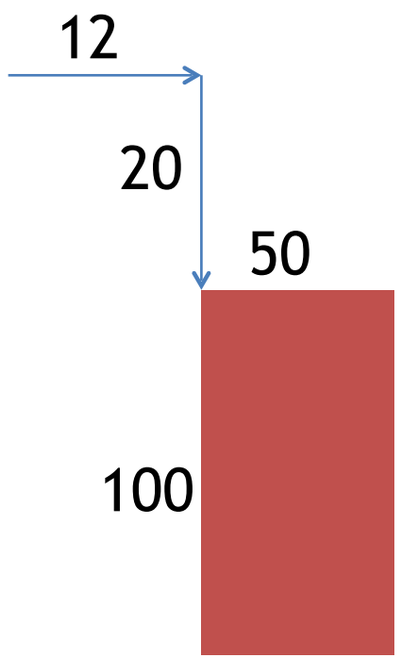
Principais Módulos

- **Draw** – Usado para desenhar
- **Image** – Manipulação de imagens do sistema
- **Mixer.Sound** – Sons simples, efeitos sonoros
- **Mixer.Music** – Player de músicas
- **Sprite** – Objetos de colisão, ex: personagens
- **Time** – Manipulação do tempo no jogo
- **Font** – Criar textos e renderizar em imagens

Desenhando um Retângulo

```
                    #rect(tela     , cor         , (X, Y, width, height))
pygame.draw.rect(screen, fireColor, (12, 20, 50, 100))
```

Note que X,Y representa a posição inicial a partir do canto superior esquerdo de onde se inicia o desenho:



Exemplo

pygameDrawing.py

```python
#importa os módulos necessários
import pygame
from pygame.locals import *
from pygame.color import THECOLORS as COR

pygame.init()
screenSize = (640, 480)
screen = pygame.display.set_mode(screenSize)
screen.fill((255, 255, 255))
pygame.display.set_caption("Drawing Examples")
clock = pygame.time.Clock()
running = True
print(COR)
while running:
    r = pygame.draw.rect(screen, COR['black'], (12, 20, 50, 100))
    pygame.draw.polygon(screen, COR['blue'], [(100,100), (100,200), (200,100), (200,200)])
    pygame.draw.circle(screen, COR['green'], (150, 300), 50)
    pygame.draw.ellipse(screen, COR['cyan'], r)
    pygame.draw.line(screen, COR['hotpink2'], (300, 350), (500, 300), 10)
    pygame.draw.lines(screen, COR['orangered1'], False, [(600, 400), (500, 300), (100, 400), (100, 200)])

    time_passed = clock.tick(30)
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False
        if (event.type == KEYUP and event.key == K_ESCAPE):
            running = False
    pygame.display.update()

pygame.quit()

```

Carregando uma Imagem

Formatos suportados: BMP, TGA, GIF (não animado), JPEG, PNG, TGA, etc.

pygameImages.py

```
#importa os módulos necessários
import pygame
from pygame.locals import *

pygame.init()
screenSize = (640, 480)
screen = pygame.display.set_mode(screenSize)
screen.fill((255, 255, 255))
pygame.display.set_caption("Loading Images Example")
clock = pygame.time.Clock()
running = True
while running:
    # Carrega uma imagem para o Pygame.
    # OBS: Carrega, mas NÃO desenha ela na tela!
    img_gif = pygame.image.load('imagens/imagem.gif').convert()
    img_png = pygame.image.load('imagens/imagem.png').convert_alpha()
    #desenha imagem
    screen.blit(img_gif, (10, 10))
    screen.blit(img_png, (300, 300))

    time_passed = clock.tick(30)
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False
        if (event.type == KEYUP and event.key == K_ESCAPE):
            running = False
    pygame.display.update()

pygame.quit()

```

Usando Fontes em Textos

- Permite renderizar TrueType Fonts (*.TTF) em imagens para o jogo

- Permite a utilização de fontes extras, especiais

pygame.font.Font(‘nome_arquivo')

pygameImages.py ×

```python
#importa os módulos necessários
import pygame
from pygame.locals import *

pygame.init()
screenSize = (640, 480)
screen = pygame.display.set_mode(screenSize)
screen.fill((255, 255, 255))
pygame.display.set_caption("Using Font Example")
clock = pygame.time.Clock()
running = True

# Carrega um arquivo de fonte para o Pygame.
fonte = pygame.font.Font('TheGodfather.ttf', 90)
while running:
    # Cria texto com a fonte carregada
    screen.blit(fonte.render('Ola mundo', True, (255, 0, 0)), (200, 150))


pyg
```

Efeitos Sonoros

- Tocando uma música uma vez:

```
pygame.mixer.music.load('foo.mp3')
pygame.mixer.music.play(0)
```

- Tocando uma música indefinidamente:

```
pygame.mixer.music.load('foo.mp3')
pygame.mixer.music.play(-1)
```

- Adicionando uma música na fila de músicas:

```
pygame.mixer.music.queue('next_song.mp3')
```

- Parando a música atual:

```
pygame.mixer.music.stop()
```

https://www.zamzar.com/convert/mp3-to-ogg/

Efeitos Sonoros

- Fazer algo quando uma música termina:

```
...

SONG_END = pygame.USEREVENT + 1

pygame.mixer.music.set_endevent(SONG_END)
pygame.mixer.music.load('song.mp3')
pygame.mixer.music.play()


...

while True:
    ...
    for event in pygame.event.get():
        ...
        if event.type == SONG_END:
            print("the song ended!")
    ...
```

Para evitar que o evento de terminar a música seja igual ao número de outro evento.

Personagens (Sprite)

- Sprite é o módulo usado para os objetos, personagens, e cenário do jogo

- Um Sprite é a forma de representar um item do jogo. Possui uma posição (na tela) e uma imagem:
  - **Sprite.rect**
  - **Sprite.image**
- As funções do módulo sprite lidam com objetos Sprite: **pygame.sprite.Sprite()**

Sprite: Detecção de Colisões

- Várias funcões do módulo sprite são para detecção de colisão.

pygame.sprite.collide_mask(sprite_a, sprite_b)

- Retorna um boolean indicando se as imagens dos dois sprites estão se sobrepondo (colidindo)

```
# Verifica se dois Sprites estão colidindo.
# Neste ponto devem existir dois sprites, bola e parede.
if pygame.sprite.collide_mask(bola, parede):
        som_colisao.play(1)
        # mude a direção da bola ou
        # faça o jogador perder uma vida e recomece o jogo
```

Sprite: Detecção de Colisões

- Detecção de colisão entre dois sprites, usando rect (posicionamento e dimensões) :

pygame.sprite.collide_rect(sprite_a, sprite_b)

- Detecção de colisão usando áreas circulares:

pygame.sprite.collide_circle(sprite_a, sprite_b)

- Outras fuções do módulo sprite, permitem outros tipos de verificação (por grupos de sprites, por camada)

Time

- Módulo responsável pela informação sobre o tempo no jogo.
- Muito útil em jogos que possuem movimentos e física
- Regula a movimentação dos objetos na tela
- Evita diferença de execução entre máquinas diferentes

TIME

```
# Cria um relógio
clock = pygame.time.Clock()
…
x , y = (10, 5)
vel_x, vel_y = 7, 3
…
While True:
        delta_tempo =
clock.tick()
        …
        x += vel_x * delta_tempo
        y += vel_y * delta_tempo

# S = S0 + v * t
```

- **clock = pygame.time.Clock()**
  - Cria um relógio que serve para se obter a variação de tempo ao longo do programa

- **clock.tick()**
  - Retorna o tempo, em milisegundos, desde a última chamada deste método
  - Deve ser uma vez por loop

Exercícios

1. **Desenhe uma Smiley Face através dos métodos que permitem desenhar no PyGame.**

2. **Crie uma playlist que permite que o usuário troque de música a partir do teclado (ex: sempre que pressionar a tecla S, muda a música).**

3. **Leia o conteúdo de um arquivo texto e escreva ele na tela do PyGame.**