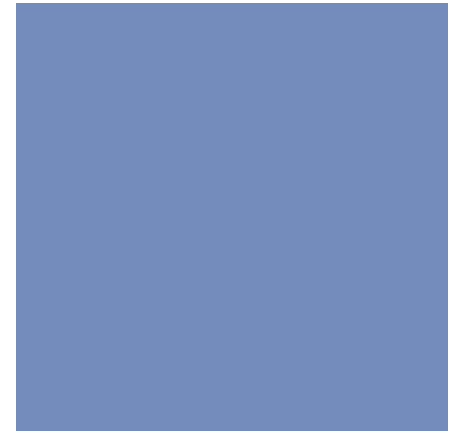
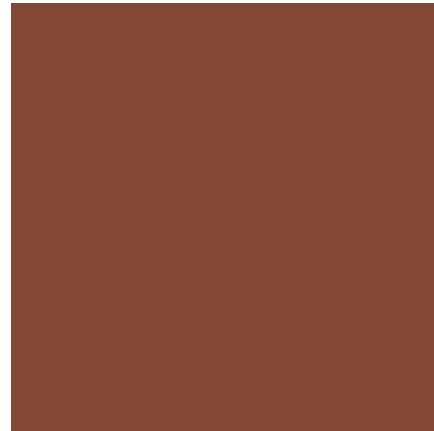




SFML

soluções tutorial



INF110 – Programação I

Prof. André Gustavo
DPI/UFV – 2025/1





SFML



- Soluções das tarefas do tutorial
- **NÃO LEIA** antes de tentar fazer as tarefas!



Window – 1-window.cpp



- Escrever uma mensagem logo antes de fechar a janela

```
sf::Event event;
while (window.pollEvent(event)) {
    // evento "fechar" acionado: fecha a janela
    if (event.type == sf::Event::Closed) {
        cout << "Fechou!\n";
        window.close();
    }
}
```

+ Event – 2b-event.cpp

- Fechar a janela quando se pressiona Shift + Esc

```
//Se pressionou Shift + ESC  
if (event.key.code == sf::Keyboard::Escape && event.key.shift)  
    window.close();
```

no tratamento de eventos



Draw shape – 4-shape.cpp

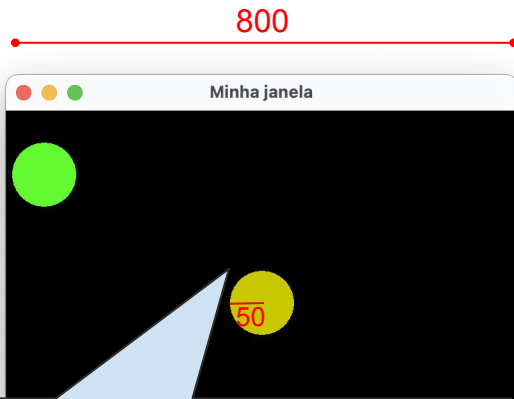
- Adicionar um círculo amarelo exatamente no centro da janela

```
// cria um círculo amarelo e define sua posicao no centro da janela  
sf::CircleShape circ2(50);  
circ2.setPosition(350, 250);  
circ2.setFillColor(sf::Color(200, 200, 00));
```

antes do loop principal

```
// desenha o círculo na janela  
window.draw(circ);  
window.draw(circ2);
```

dentro do loop principal



setPosition é em relação ao canto superior esquerdo da "caixa" que envolve o shape



Draw shape – 4-shape.cpp



- Adicionar um retângulo 100 x 50 em cada canto

```
// cria um retângulo azul 100x50  
sf::RectangleShape ret(sf::Vector2f(100,50));  
ret.setFillColor(sf::Color::Blue);
```

antes do loop principal

```
ret.setPosition(0,0);  
window.draw(ret);  
ret.setPosition(700,0);  
window.draw(ret);  
ret.setPosition(0,550);  
window.draw(ret);  
ret.setPosition(700,550);  
window.draw(ret);
```

dentro do loop principal



Draw shape – 4-shape.cpp



- Mudar a cor do círculo aleatoriamente

```
// desenha o círculo na janela  
int r = rand()%256;  
int g = rand()%256;  
int b = rand()%256;  
circ.setFillColor(sf::Color(r,g,b));  
window.draw(circ);
```



Draw shape – 4b-shape.cpp

- Movimentar o quadrado também para cima e para baixo

```
// tecla pressionada
if (event.type == sf::Event::KeyPressed) {
    if (event.key.code == sf::Keyboard::Left)
        posX -= 10;    // left key: move o quadrado para esquerda
    else if (event.key.code == sf::Keyboard::Right)
        posX += 10;    // right key: move o quadrado para direita
    else if (event.key.code == sf::Keyboard::Up)
        posY -= 10;    // up key: move o quadrado para cima
    else if (event.key.code == sf::Keyboard::Down)
        posY += 10;    // down key: move o quadrado para baixo
}
```




Draw shape – 4b-shape.cpp

- Impedir que o quadrado saia da janela

```
// tecla pressionada
if (event.type == sf::Event::KeyPressed) {
    if (event.key.code == sf::Keyboard::Left)
        posX -= 10; // left key: move o quadrado para esquerda
    else if (event.key.code == sf::Keyboard::Right)
        posX += 10; // right key: move o quadrado para direita
    else if (event.key.code == sf::Keyboard::Up)
        posY -= 10; // up key: move o quadrado para cima
    else if (event.key.code == sf::Keyboard::Down)
        posY += 10; // down key: move o quadrado para baixo
}
```

```
if (posx < 0) posX = 0;
if (posx > 750) posX = 750;
if (posy < 0) posY = 0;
if (posy > 550) posY = 550;
```

não tem problema posX ficar negativo e ser corrigido aqui; o quadrado só é desenhado depois, no window.draw



Draw shape – 4b-shape.cpp



- Não impedir que saia da janela, fazê-lo aparecer do outro lado

```
if (posx<-50) posx+=850;  
if (posx>800) posx-=850;  
if (posy<-50) posy+=650;  
if (posy>600) posy-=650;
```

deixei uma margem para ele sumir
e reaparecer do outro lado



Time – 6-time.cpp

- Mover a vida na mesma velocidade, mas de forma mais suave

```
if (clock.getElapsedTime() > sf::seconds(0.01)) { // tempo desde  
    clock.restart();           // recomeça contagem do tempo  
    posxup += 1;               // muda posição de desenho da vida  
}
```

diminuir o tempo de atualização

diminuir o tamanho do passo



Time – 6-time.cpp



- Não deixar a vida escapar da tela!
Quando atingir o limite da direita, passar a movimentá-la para a esquerda

Tentativa #1 (não funciona!!)

```
// Muda a posição do sprite vida a cada 0.01 segundos
if (clock.getElapsedTime() > sf::seconds(0.01)) { // tempo desde
    clock.restart(); // recomeça contagem do tempo
    posxup += 1;
}
if (posxup > 752)
    posxup -= 1;
```

a figura tem tamanho 48x48

quando passar do limite, mover para esquerda

quando passa do limite, volta; mas na iteração seguinte vai de novo, fica indo e voltando



Time – 6-time.cpp



- Não deixar a vida escapar da tela!
Quando atingir o limite da direita, passar a movimentá-la para a esquerda

Tentativa #2 (não funciona!!)

```
// Muda a posição do sprite vida a cada 0.01 segundos
if (clock.getElapsedTime() > sf::seconds(0.01)) { // tempo desde
    clock.restart();           // recomeça contagem do tempo
    posxup += 1;               // muda posição de desenho da vida
}
if (posxup > 752) {
    while (posxup > 0)
        posxup -= 1;
}
```

quando passar do limite, fica
movendo para esquerda

fica no while até acabar, sem redesenhar... volta para esquerda, mas não mostra voltando
via de regra, não use repetição aqui!! deixe o loop clear/draw/display redesenhar!



Time – 6-time.cpp

- Não deixar a vida escapar da tela!
Quando atingir o limite da direita, passar a movimentá-la para a esquerda

Tentativa #3 (ok!! 🙌)

booleano para sentido da direção

```
float posxup = 500, posyup = 50; // posição inicial do sprite de vida
bool movedireita = true; // está movendo para a direita?
```

```
// Muda a posição do sprite vida a cada 0.01 segundos
if (clock.getElapsedTime() > sf::seconds(0.01)) { // tempo desde último
    clock.restart(); // recomeça contagem do tempo

    if (movedireita) // muda posição de desenho da vida
        posxup += 1; // de acordo com a direção do movimento
    else
        posxup -= 1;

    if (posxup >= 752)
        movedireita = false;
    if (posxup <= 0)
        movedireita = true;
}
```

move de acordo com o booleano

ajusta o sentido, se necessário



Juntando tudo! – 7-pacman.cpp

- Não deixar o Pac-Man atravessar as paredes

```
// tecla pressionada
if (event.type == sf::Event::KeyPressed) {
    if (event.key.code == sf::Keyboard::Left && mapa[posy][posx-1] != '1')
        posX--; // left key: move o PacMan para esquerda
    else if (event.key.code == sf::Keyboard::Right && mapa[posy][posx+1] != '1')
        posX++; // right key: move o PacMan para direita
```

verifica se a posição que ele vai não é uma parede

Muito cuidado porque em coordenadas (x,y), x é horizontal (portanto, coluna) e y é vertical (linha). Coordenadas de tela são o contrário da matriz!!



Juntando tudo! – 7-pacman.cpp

- Mudar a figura do Pac-Man quando ele se move para esquerda

```
sf::Texture texture2;  
if (!texture2.loadFromFile("pacman-esq.png")) {  
    std::cout << "Erro lendo imagem pacman.png\n";  
    return 0;  
}  
sf::Sprite spriteesq;  
spriteesq.setTexture(texture2);  
  
bool moveesquerda = false;
```

criar sprite com pacman
olhando para esquerda

booleano para direção

```
// tecla pressionada  
if (event.type == sf::Event::KeyPressed) {  
    if (event.key.code == sf::Keyboard::Left && mapa[posy][posx-1]!='1') {  
        posx--; // left key: move o PacMan para esquerda  
        moveesquerda = true;  
    }  
    else if (event.key.code == sf::Keyboard::Right && mapa[posy][posx+1]!='1') {  
        posx++; // right key: move o PacMan para direita  
        moveesquerda = false;  
    }  
}
```

muda booleano de direção

```
// desenha PacMan  
if (moveesquerda) {  
    spriteesq.setPosition(posx*50, posy*50);  
    window.draw(spriteesq);  
}  
else {  
    sprite.setPosition(posx*50, posy*50);  
    window.draw(sprite);  
}
```

desenha de acordo com a
direção de movimento



Movimento contínuo – 7b-pacman.cpp

- Adicionar opção de reiniciar: tecla 'R' reposiciona na posição inicial

```
int initposx = 7; // posicao inicial do PacMan  
int initposy = 6;
```

antes do loop do jogo

```
}  
else if (event.key.code == sf::Keyboard::R) {  
    posx = initposx;  
    posy = initposy;  
}
```

no tratamento de eventos



Dica: use e abuse de booleanos!



- Booleanos para sentido de movimento do personagem
 - ativar/desativar booleanos quando pressiona tecla
 - mudar posição de acordo com booleanos
 - desenhar figura certa de acordo com booleanos
- Booleano para jogo ativo/terminado
- Booleano para frutinha ativada/desativada
- Booleanos/contadores para múltiplas figuras
 - abrir/fechar boca do PacMan
 - mover pernas de personagens
- ...

