

Introdução a Programação de Jogos

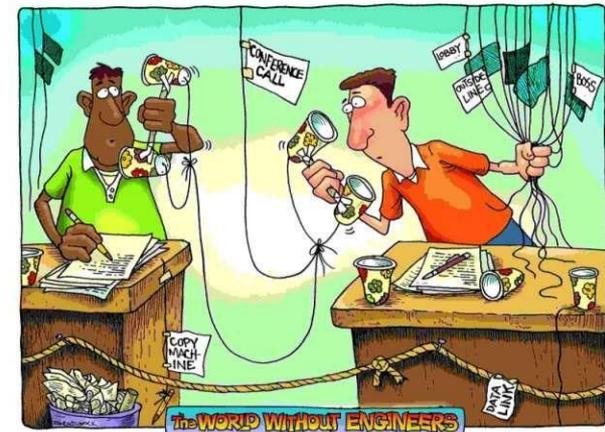
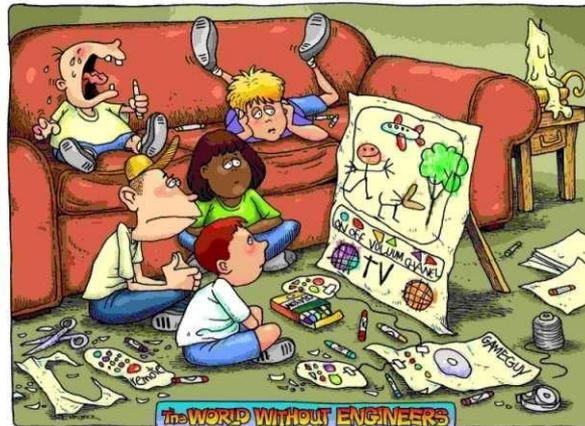
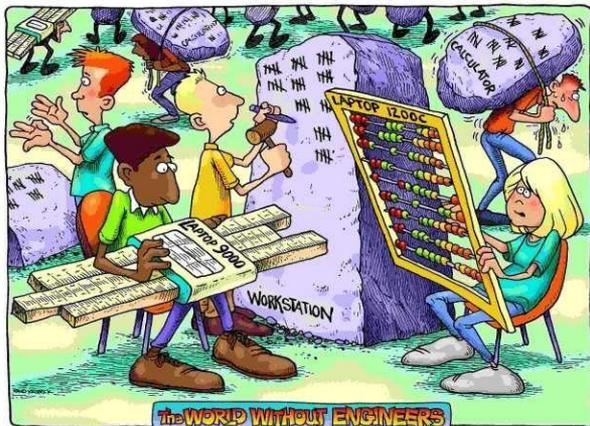
Aula 09 – Projeto de Jogos Eletrônicos

Prof. Augusto Baffa

< abaffa@inf.puc-rio.br >

Engenharia da Computação

- O Engenheiro de Computação é um profissional de formação generalista, que atua em diversas áreas da computação:
 - Projeta, desenvolve e implementa equipamentos e dispositivos computacionais, periféricos e sistemas que integram hardware e software;
 - Especifica, desenvolve, implementa, adapta, industrializa, instala e mantém sistemas computacionais;



Jogos Eletrônicos

- **O que é um jogo?**
 - Jogar uma bola contra uma parede pode ser considerado um jogo?



Jogos Eletrônicos

- **E se nós adicionarmos regras?**
 - Jogar a bola com a mão direita e pega-la com a mão esquerda;
 - Nunca deixar a bola cair.
- **Podemos adicionar também condições de vitória e derrota:**
 - O jogador ganha ao jogar e pegar a bola 10 vezes;
 - O jogador perde se violar alguma das regras.

- **E agora, temos um jogo?**



Jogos Eletrônicos

- **O que é um jogo?**

- Qualquer atividade em que exista pelo menos um jogador (como indivíduo praticante do jogo);
- Devem existir regras;
- Devem existir objetivos ou condições de vitória;
- Deve ser uma forma de entretenimento.

- **O que é um jogo eletrônico?**

- Qualquer jogo que é jogado em uma tela de vídeo.



Oficina de Programação

- **Jogo simples: Space Shooter**

1. Controle da nave pelo teclado;
2. Mantendo a nave na tela;
3. Geração de estrelas;
4. Turbo para a nave;
5. Controle da nave com Arduino;
6. Atirar com a nave;

- **Extra:**

6. Gerar inimigos;
7. Detectar colisão (nave/inimigos);
8. Detectar colisão (tiros/inimigos);
9. Controle de vidas e pontuação;



Projeto: Space Shooter

- **Programa Base:**

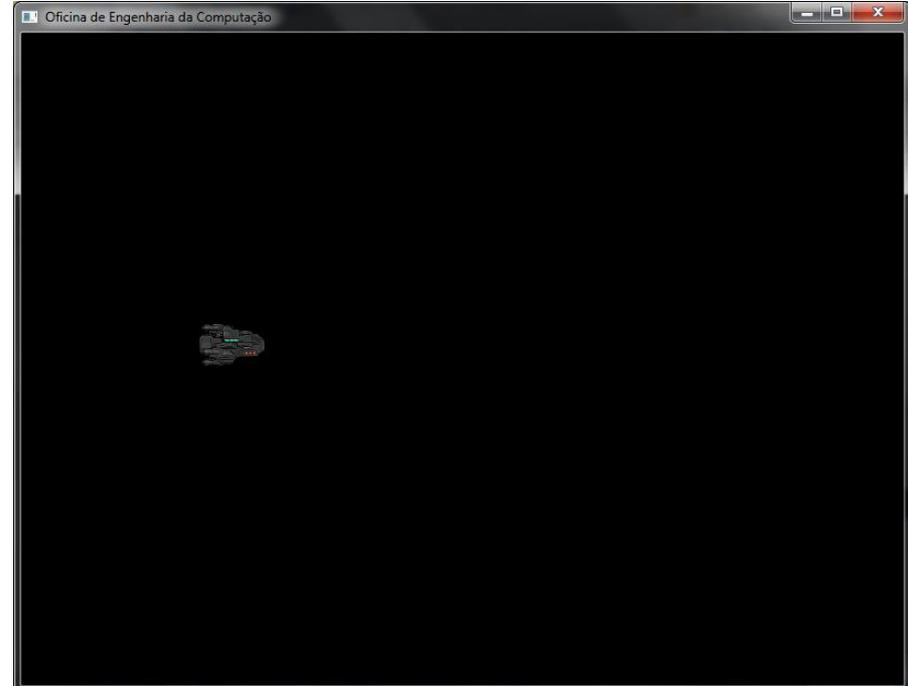
- <http://www.inf.puc-rio.br/~abaffa/iue1503/>
 - Compilando e executando;
 - Lógica do loop principal;
 - Lógica da movimentação da nave;

```
//funcao de loop principal do jogo
void MainLoop()
{
    if (key_states[KEY_RIGHT] == true) //se a tela da direita for pressionada
    {
        //movimenta a nava para frente
        player.pos_x = player.pos_x + (player.velocidade * graphics.GetElapsedTime());
    }

    //desenha a nave na posicao pos_x e pos_y da tela
    graphics.DrawImage2D(player.pos_x, player.pos_y, 128, 128, player.imagemNormal);
}

//Funcao para verificar o pressionamento de teclas do teclado
void KeyboardInput(int key, int state, int x, int y)
{
    if (state == KEY_STATE_DOWN)
        key_states[key] = true;
    else if (state == KEY_STATE_UP)
        key_states[key] = false;
}

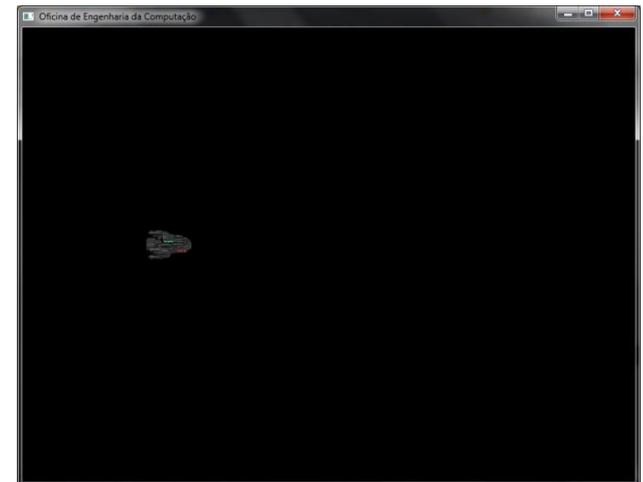
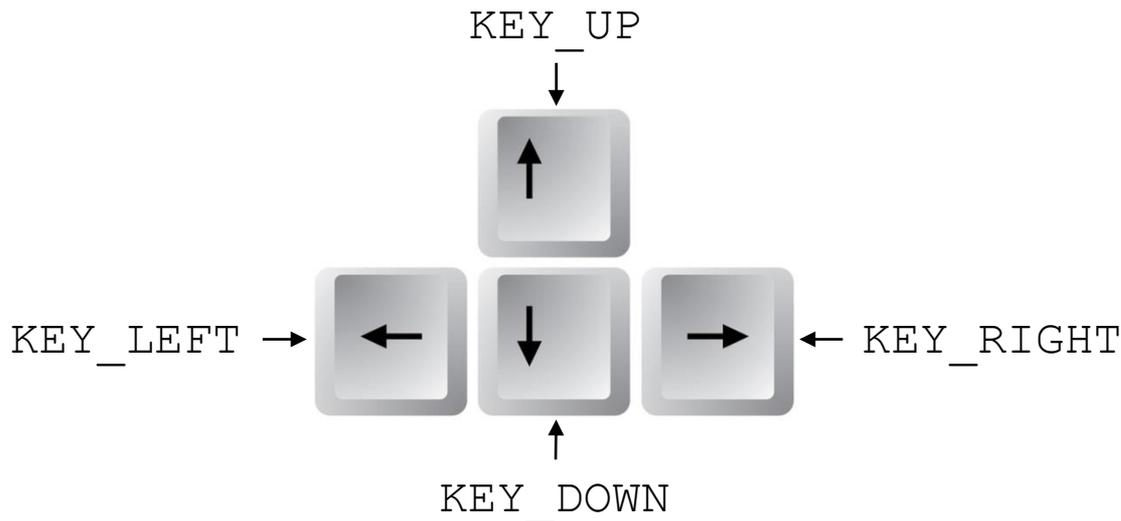
int main(void)
{
    //cria janela para o jogo
    graphics.CreateMainwindow(800, 600, "Oficina de Engenharia da Computação");
    //define a cor de fundo da tela: preto (RGB - 0,0,0)
    graphics.SetBackgroundColor(0, 0, 0);
}
```



Projeto: Space Shooter

- **Tarefa 01:** Fazer a nave movimentar-se em todas as direções usando as setas direcionais do teclado.

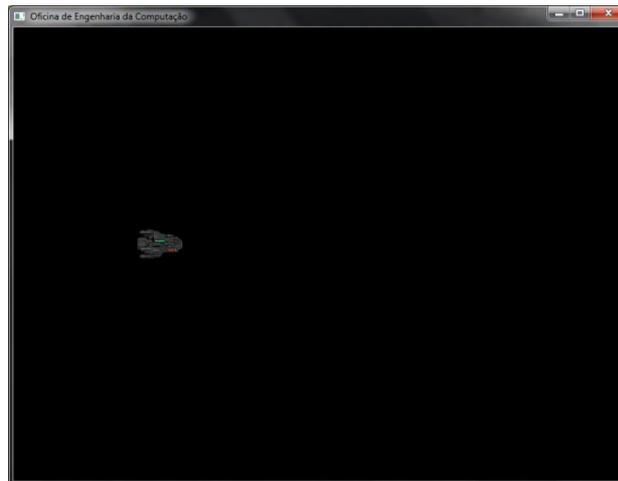
```
if (key_states[KEY_RIGHT] == true) //se a seta da direita for pressionada
{
    //movimenta a nave para frente
    player.pos_x = player.pos_x + (player.velocidade * graphics.GetElapsedTime());
}
...
```



Projeto: Space Shooter

- **Tarefa 02:** Evitar que a nave saia dos limites da tela.

```
if (key_states[KEY_RIGHT] == true) //se a seta da direita for pressionada
{
    //movimenta a nave para frente se ela não estiver fora da tela
    if (player.pos_x + player.imagemNormal.width < graphics.GetScreenWidth())
        player.pos_x = player.pos_x + (player.velocidade * graphics.GetElapsedTime());
}
...
```

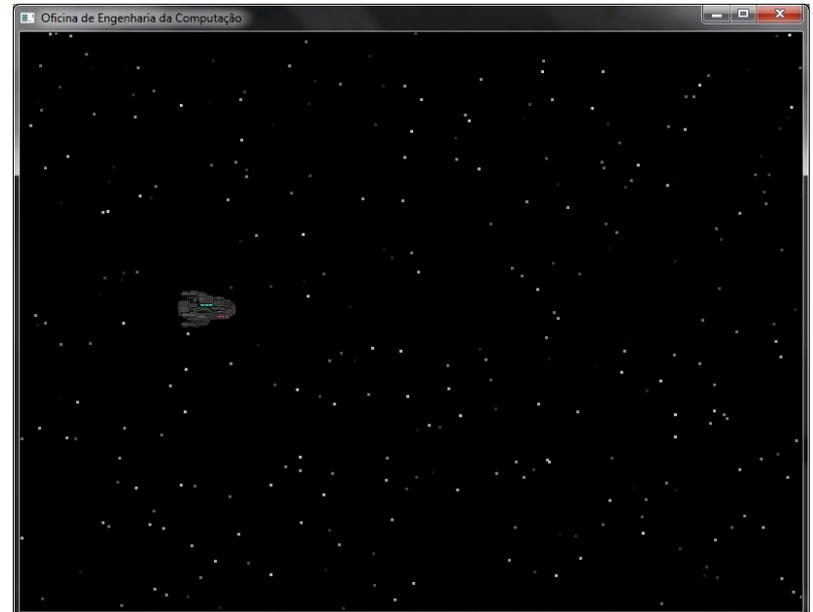


Projeto: Space Shooter

- **Tarefa 03:** Gerar estrelas aleatoriamente para simular o espaço.
 - Definir a estrutura e o vetor de estrelas;
 - Gerar a posição e velocidade das estrelas;
 - Desenhar e movimentar as estrelas;

```
//estrutura de informações das estrelas
typedef struct estrela{
    float pos_x;
    float pos_y;
    float velocidade;
} Estrela;

//vetor com 300 estrelas
Estrela estrelas[300];
...
```



Projeto: Space Shooter

- **Tarefa 04:** Ativar o turbo para a nave ao pressionar as teclas shift.



Sem Turbo



Com Turbo

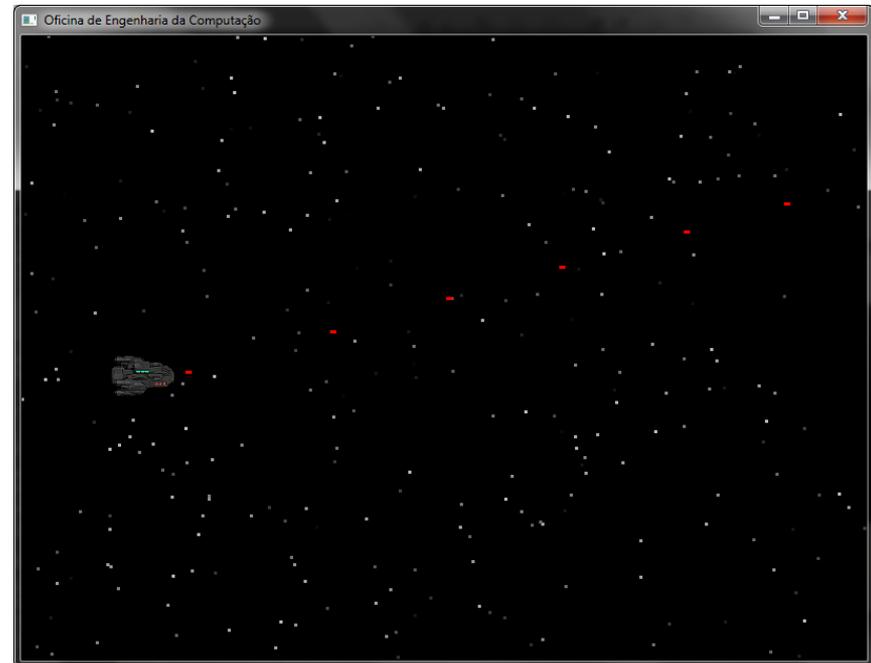
```
...
if (player.turbo == true) //desenha a nave na tela com o turbo ativo
    graphics.DrawImage2D(player.pos_x, player.pos_y, 128, 128, player.imagemTurbo);
else //desenha a nave na tela sem o turbo
    graphics.DrawImage2D(player.pos_x, player.pos_y, 128, 128, player.imagemNormal);
...
```

Projeto: Space Shooter

- **Tarefa 05:** Atirar com a nave ao pressionar a tecla espaço.

```
//estrutura de informacoes do tiro
typedef struct tiro{
    float pos_x;
    float pos_y;
    bool ativo;
}Tiro;
```

```
//estrutura de informacoes da nave
typedef struct nave{
    .
    .
    .
    Tiro tiros[20];
    int tiro_cont;
    bool atirar;
} Nave;
```



Projeto: Space Shooter

- **Tarefas Complementares:**

a) Sistema de geração de inimigos aleatórios;



b) Método de detecção de colisão entre a nave e os inimigos;

c) Método de detecção de colisão entre os tiros disparados e os inimigos;

d) Sistema para controlar o número de vidas e a pontuação;