

Introdução à Engenharia

ENG1000

Aula 14 – Estruturas e Sprites

Prof. Augusto Baffa

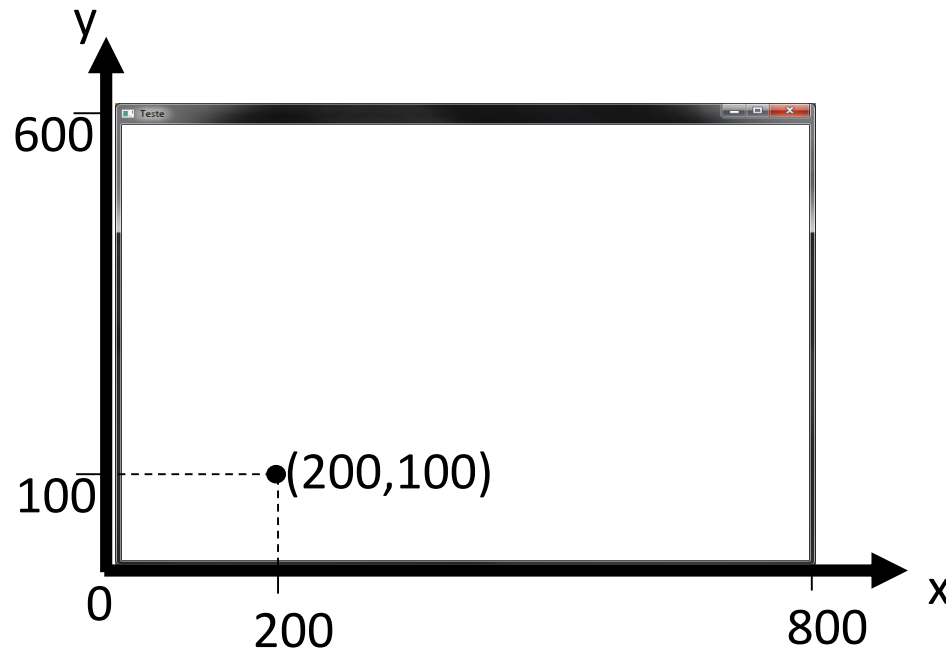
< abaffa@inf.puc-rio.br >

Estruturas

Struct são coleções de dados heterogêneos agrupados em uma mesma estrutura de Dados

Exemplo:

Armazenar as coordenadas (x,y) de um ponto

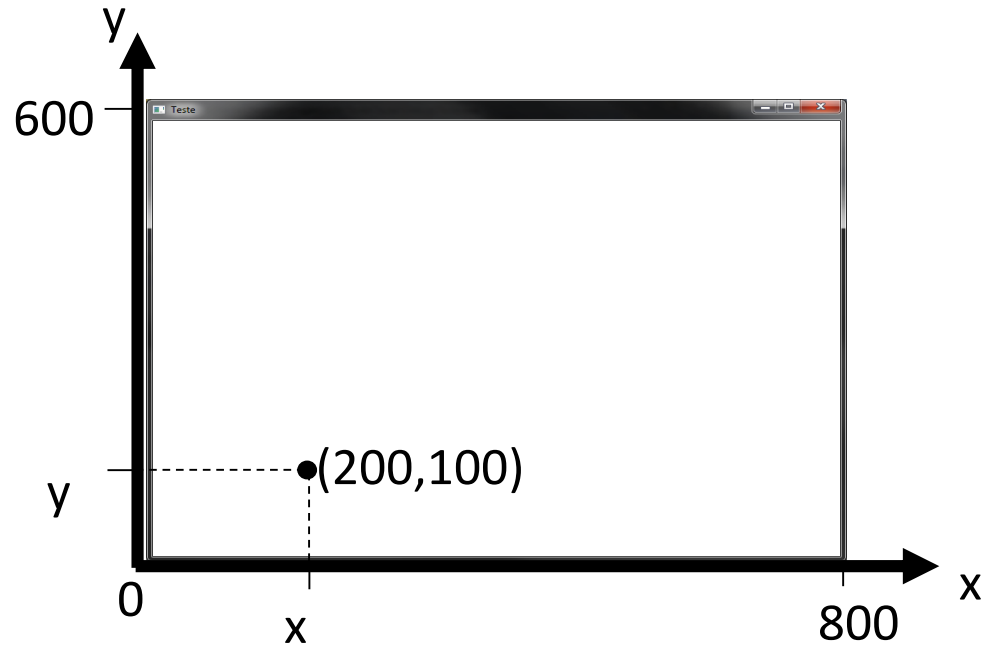


Estruturas

- Declaração:

```
struct coordenada{  
    int x;  
    int y;  
};
```

```
coordenada p1, p2;
```



- A estrutura “coordenada” contém dois inteiros, x e y
- p1 e p2 são duas variáveis tipo coordenada.

Declaração de Estrutura

- Formato da declaração:

```
struct nome_da_estrutura {  
    tipo_1 dado_1;  
    tipo_2 dado_2;  
    ...  
    tipo_n dado_n;  
};
```

- A estrutura pode agrupar um número arbitrário de dados de tipos diferentes
- Pode-se nomear a estrutura para referenciá-la

Estruturas

- Acesso aos dados:

```
variável_struct.campo
```

- Ex:

```
p1.x = 10;
```

```
p1.y = 20;
```

```
p2.x = 15;
```

```
p2.y = 15;
```

```
if (p1.x >= p2.x) &&  
    (p1.y >= p2.y) ...
```

struct x typedef struct

- Usando Struct

```
struct struct_name{
    int one;
    int two;
};

...

void whatever(struct struct_name value);
```

- Usando Typedef Struct

```
typedef struct struct_name{
    int one;
    int two;
} typedef_name;

...

void whatever(typedef_name value);
```

Exemplo de Estrutura

- Exemplo:

```
typedef struct train
{
    float x;           //posicao x do trem
    float y;           //posicao y do trem
    int animation_clip; //clipa da animacao do trem
    int animation_index; //indice atual da animacao do trem
    float time_next_frame; //contador de tempo para avançar na animacao
    bool stoped;       //indicador de trem parado
    float speed;       //velocidade do trem
    bool light_active; //indicador de luz ativa do trem
    float angle;       //angulo do trem
    int track_sector;  //setor atual do trem na pista
} Train;

Train train;
```

Exemplo de Estrutura

- Exemplo 2:

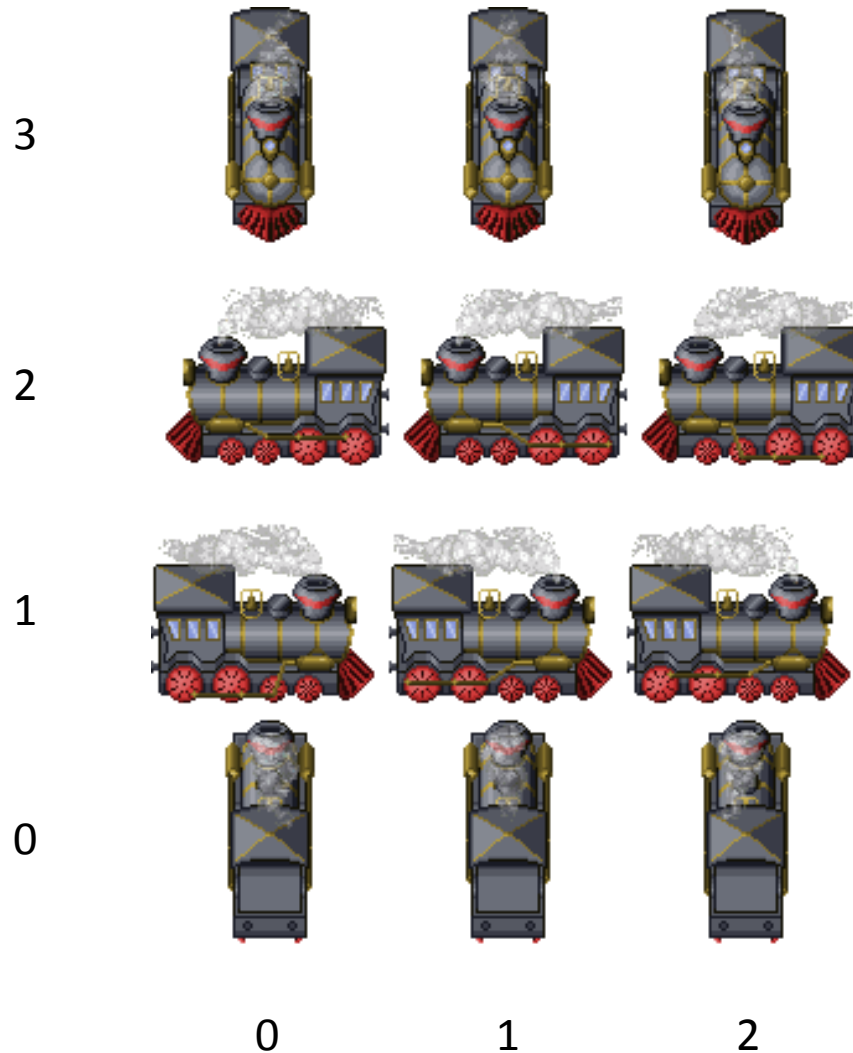
```
//Estrutura para armazenar os dados do sprite do trem
```

```
typedef struct sprite{  
    int x;  
    int y;  
    int width;  
    int height;  
} Sprite;
```

```
//Variaveis para armazenar os sprites e imagens usadas na aplicacao
```

```
Sprite sprite[4][4];
```


Carregando Sprites com Struct



Carregando Sprites com Struct

- **Exemplo de carga de Sprites:**

```
#define SPRITE_SIZE 96
```

```
Sprite sprite[4][3];  
Image sprite_sheet;
```

...

```
int main(void)  
{  
    ...  
    sprite_sheet.LoadPNGImage("Train3.png");  
    InitSprites();  
    ...  
}
```

Carregando Sprites com Struct

```
void InitSprites()  
{  
    int x = 0, y = 0;  
    int spritex = 0;  
    int spritey = 0;  
  
    for (int y = 0; y < 3; y++)  
        for (int x = 0; x < 4; x++)  
            {  
                sprite[y][x].x = 0 + SPRITE_SIZE * x;  
                sprite[y][x].y = 0 + SPRITE_SIZE * y;  
                sprite[y][x].width = SPRITE_SIZE;  
                sprite[y][x].height = SPRITE_SIZE;  
            }  
}
```

...

...

Utilizando as variáveis de Estrutura

```
int main(void)
{
    ...
    //Inicializa as variaveis de controle do trem
    train.x = 0;
    train.y = 0;
    train.animation_clip = 3;
    train.animation_index = 0;
    train.time_next_frame = 0;
    train.stoped = true;
    train.speed = 0;
    train.light_active = false;
    train.angle = -1;
    train.track_sector = 1;
    ...
}
```

Utilizando as variáveis de Estrutura

- Desenhando o Trem:

256 x 256 é o tamanho da imagem de `sprite_sheet`

```
graphics.DrawImage2D(train.x, train.y, 256, 256,  
sprite[train.animation_clip][train.animation_index].x,  
sprite[train.animation_clip][train.animation_index].y,  
sprite[train.animation_clip][train.animation_index].width,  
sprite[train.animation_clip][train.animation_index].height, sprite_sheet);
```