

playAPC: Utilização de uma Biblioteca Gráfica no Aprendizado de Algoritmos e Programação de Computadores

Alexandre Zaghetto
zaghetto@unb.br

Sinayra Moreira
sinayra@outlook.com

Mateus Mendelson
mendelsonmateus@gmail.com

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

O presente conjunto de *slides* não pode ser reutilizado ou republicado sem a permissão do autor.

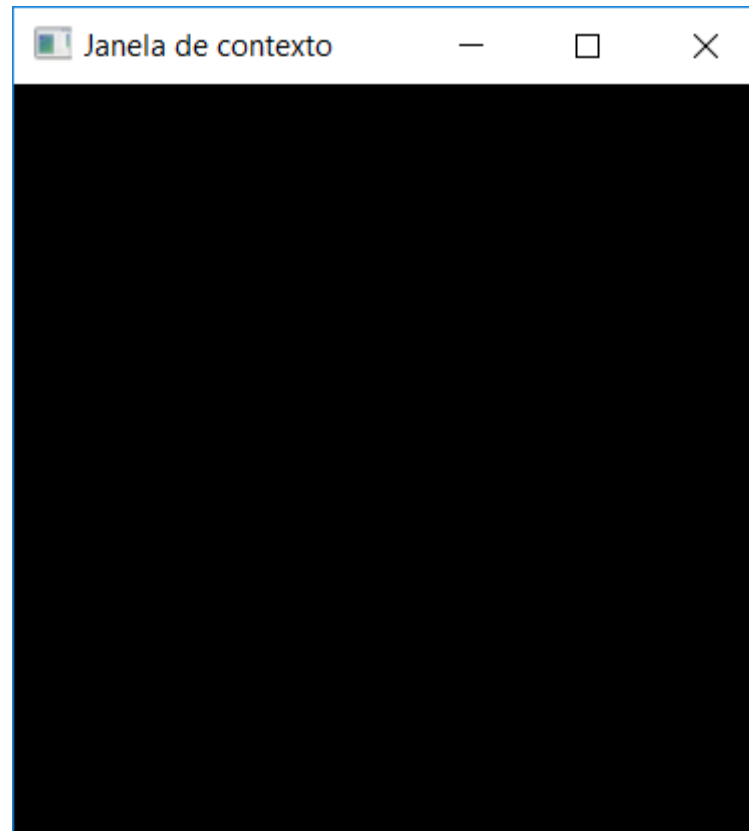
1. Instalar e Testar a playAPC

<http://playapc.zaghetto.com>



1. Instalar e Testar a playAPC

1.1 Janela de contexto



1. Instalar e Testar a playAPC

1.1 Abre uma janela de contexto

- AbreJanela()
- Desenha()



1. Instalar e Testar a playAPC

1.1 Abre uma janela de contexto

```
#include <playAPC/playapc.h>

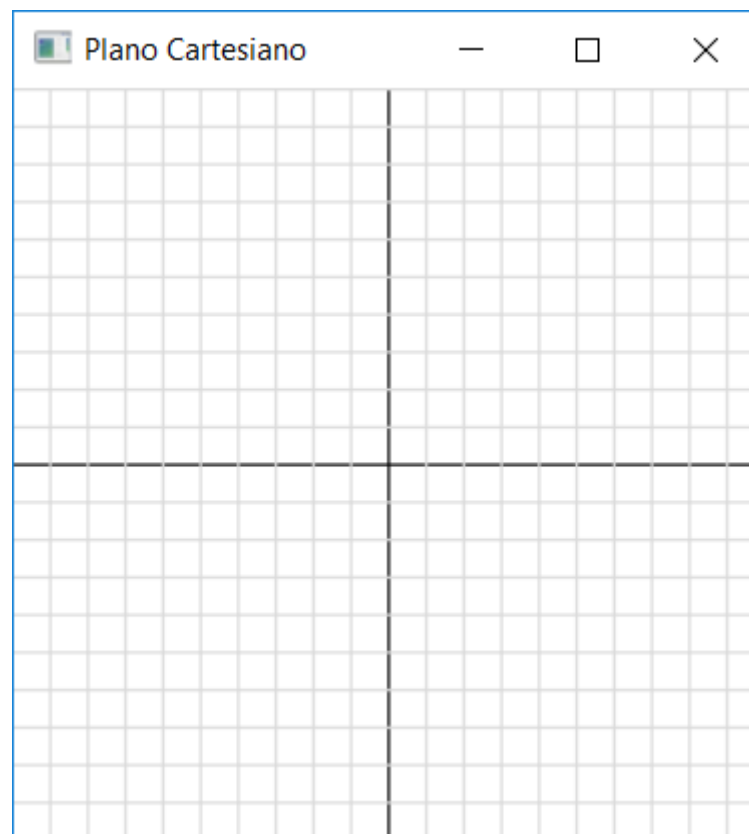
int main(){
    AbreJanela(300, 300, "Janela");

    Desenha();

    return 0;
}
```

1. Instalar e Testar a playAPC

1.2 Mostrar o plano Cartesiano



1. Instalar e Testar a playAPC

1.2 Mostrar o plano Cartesiano

- `MostraPlanoCartesiano()`

1. Instalar e Testar a playAPC

1.2 Mostrar o plano Cartesiano

```
#include <playAPC/playapc.h>

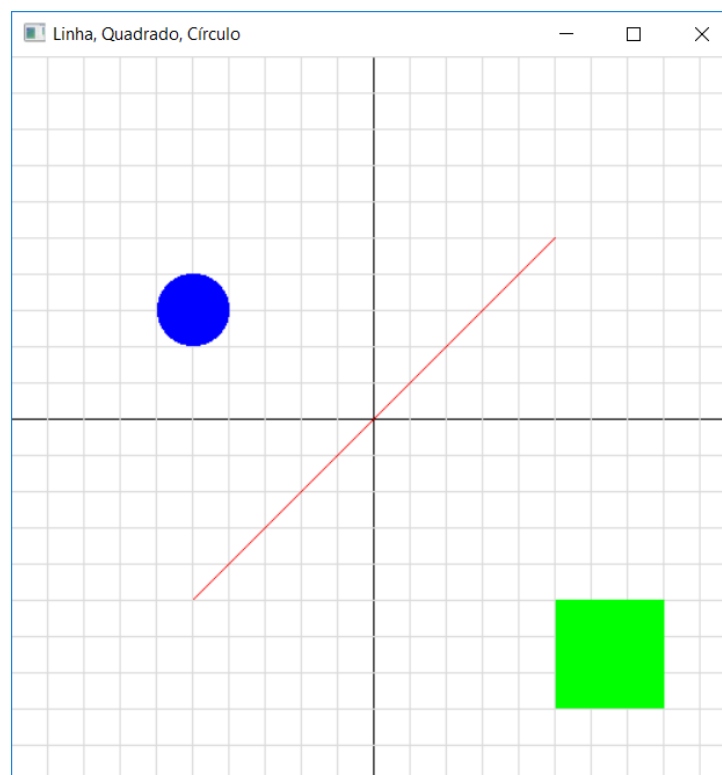
int main(){
    AbreJanela(300, 300, "Plano Cartesiano");
    PintarFundo(255, 255, 255);

    MostraPlanoCartesiano(10);
    Desenha();

    return 0;
}
```

2. Desenhar Figuras Geométricas Estáticas

2.1 Linha, Quadrado e Círculo



2. Desenhar Figuras Geométricas Estáticas

2.1 Linha, Quadrado e Círculo

- Ponto
- PintarFundo()
- CriaReta()
- CriaQuadrado()
- CriaCirculo()
- Pintar()



2. Desenhar Figuras Geométricas Estáticas

2.1 Linha, Quadrado e Círculo

```
#include <playAPC/playapc.h>
```

```
int main(){
```

```
    Ponto p1, p2;
```

```
    AbreJanela(500, 500, "Figuras Geométricas");
```

```
    PintarFundo(255, 255, 255);
```

```
    MostraPlanoCartesiano(10);
```

2. Desenhar Figuras Geométricas Estáticas

2.1 Linha, Quadrado e Círculo

```
p1.x = -50;  
p1.y = -50;  
p2.x = 50;  
p2.y = 50;  
CriaReta(p1, p2);  
Pintar(255, 0, 0);
```

```
p1.x = 50;  
p1.y = -80;  
CriaQuadrado(30, p1);  
Pintar(0, 255, 0);
```

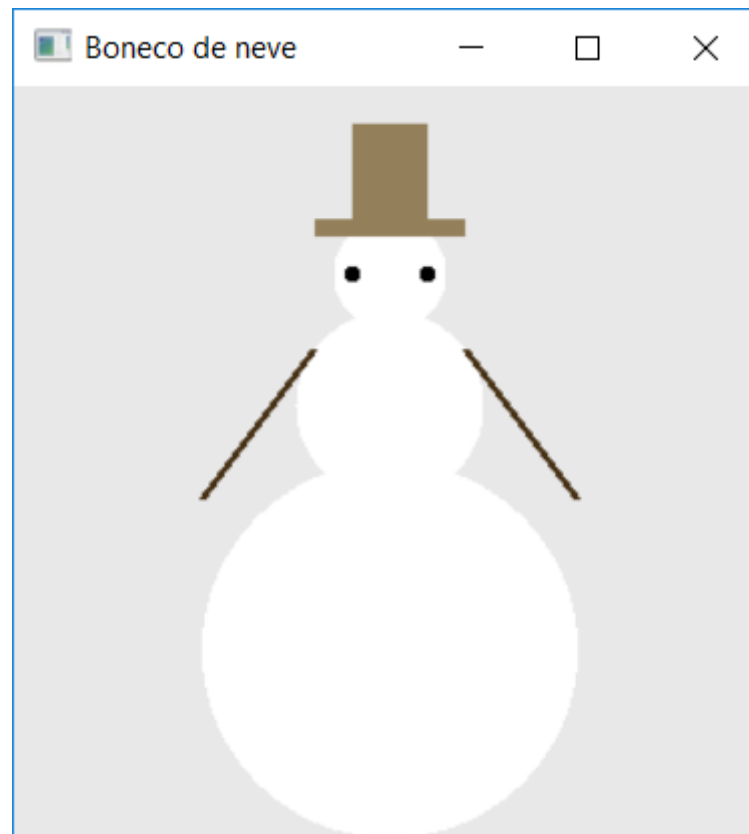
2. Desenhar Figuras Geométricas Estáticas

2.1 Linha, Quadrado e Círculo

```
p1.x = -50;  
p1.y = 30;  
CriaCirculo(10, p1);  
Pintar(0, 0, 255);  
  
Desenha();  
  
return 0;  
}
```

2. Desenhar Figuras Geométricas Estáticas

2.2 Boneco de Neve



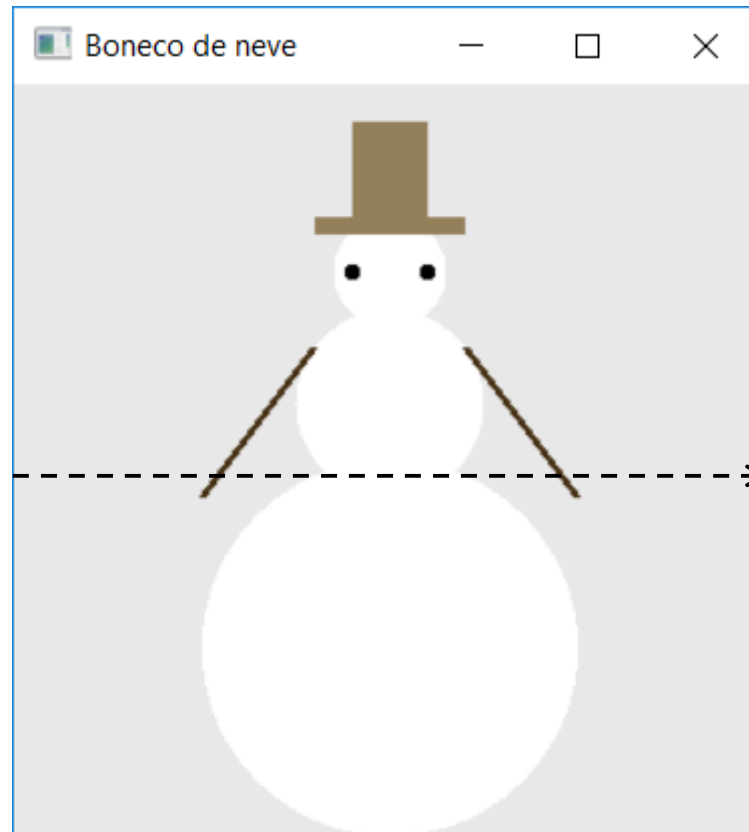
2. Desenhar Figuras Geométricas Estáticas

2.2 Boneco de Neve

- `CriaRetangulo()`
- `Grafite()`

3. Movimento

3.1 Movimentar do canto do esquerdo para o o canto direito



3. Movimento

3.1 Movimentar do canto do esquerdo para o canto direito

- `Move()`
- `Desenha1Frame()`

3. Movimento

3.1 Movimentar do canto do esquerdo para o canto direito

```
#include <playAPC/playapc.h>
```

```
int criaBonecoDeNeve(){
```

```
    int boneco;  
    boneco = CriaGrupo();
```

```
    <CONSTRUIR BONECO>
```

```
    return boneco;  
}
```

3. Movimento

3.1 Movimentar do canto do esquerdo para o canto direito

```
int main(){  
    Ponto p;  
    int boneco;  
  
    AbreJanela(300, 300, "Boneco");  
    PintarFundo(232, 232, 232);  
  
    boneco = criaBonecoDeNeve();  
  
    p.x = -100;  
    p.y = -80;
```

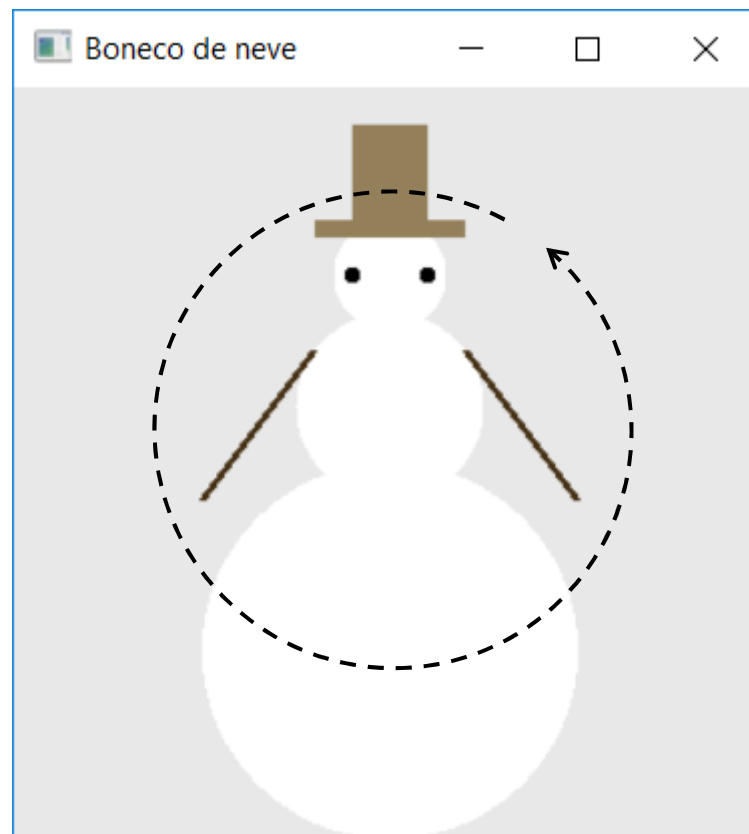
3. Movimento

3.1 Movimentar do canto do esquerdo para o canto direito

```
for(float i = -100; i < 200; i++){  
    Move(p, boneco);  
    p.x = i;  
    Desenha1Frame();  
}  
  
Desenha();  
  
return 0;  
}
```

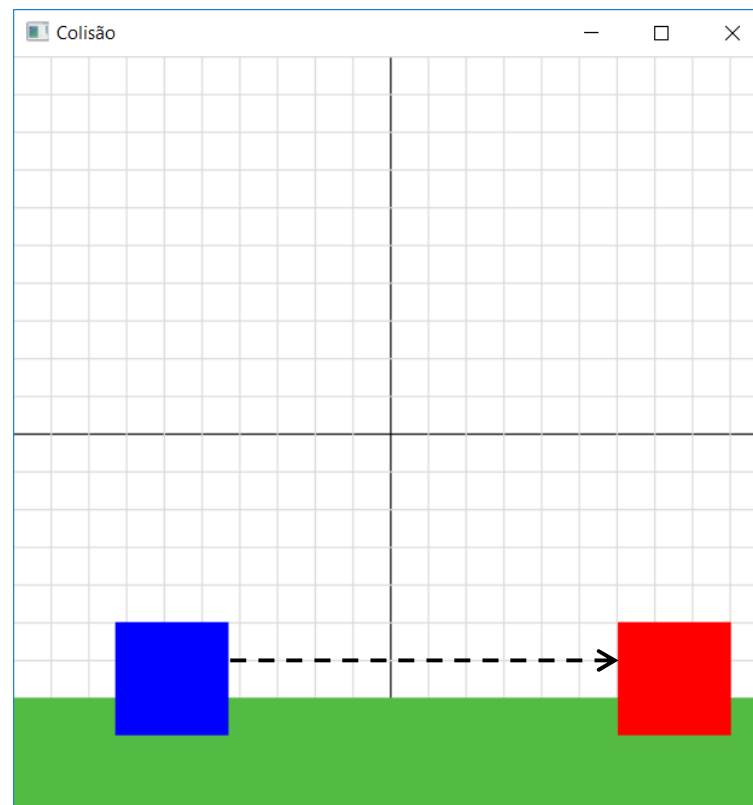
3. Movimento

3.2 Movimentar segundo uma trajetória circular



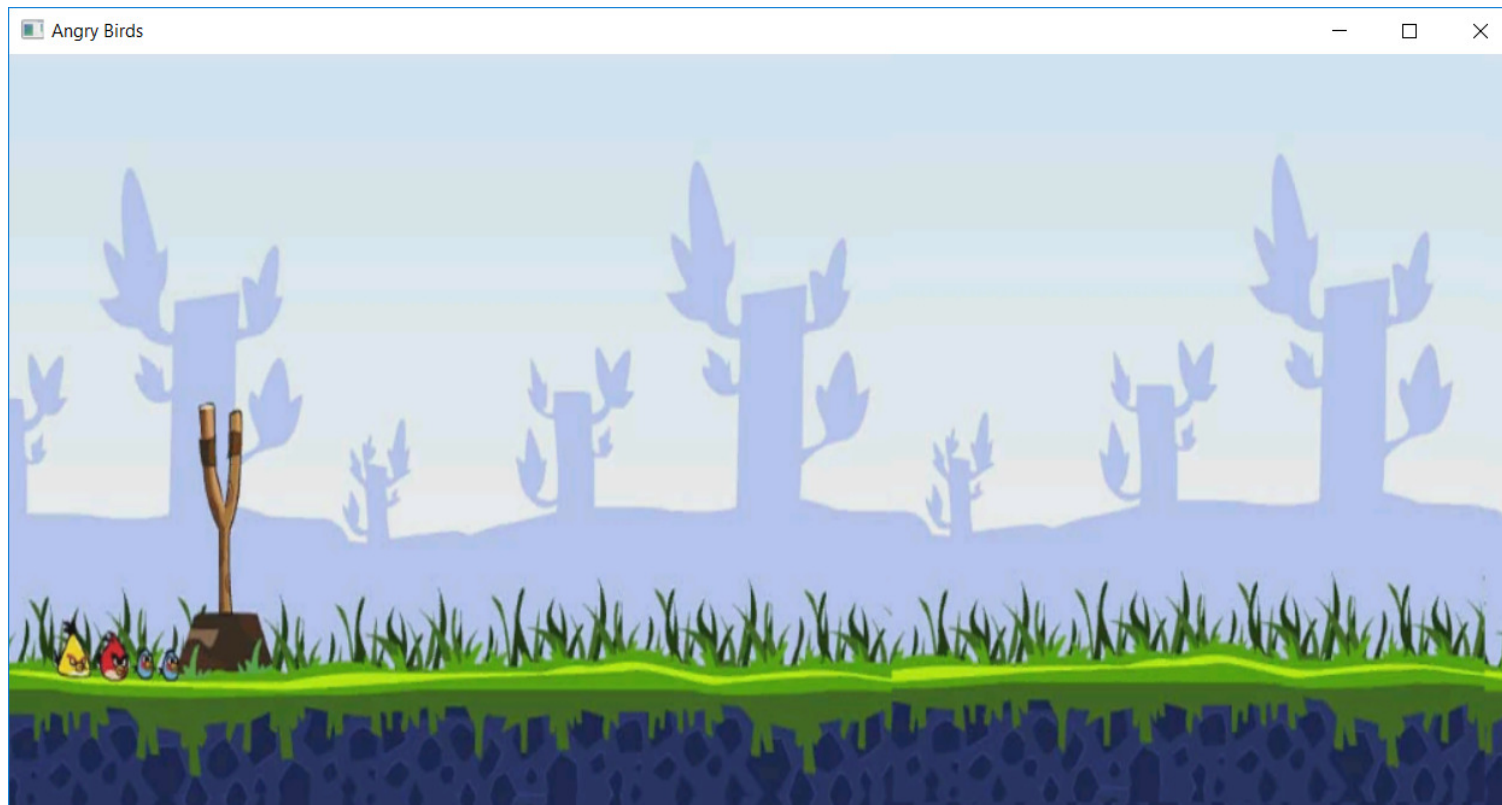
4. Colisão

4.1 Movimentar segundo uma trajetória retilínea



5. Manipulação de Imagens

5.1 Carregar um fundo



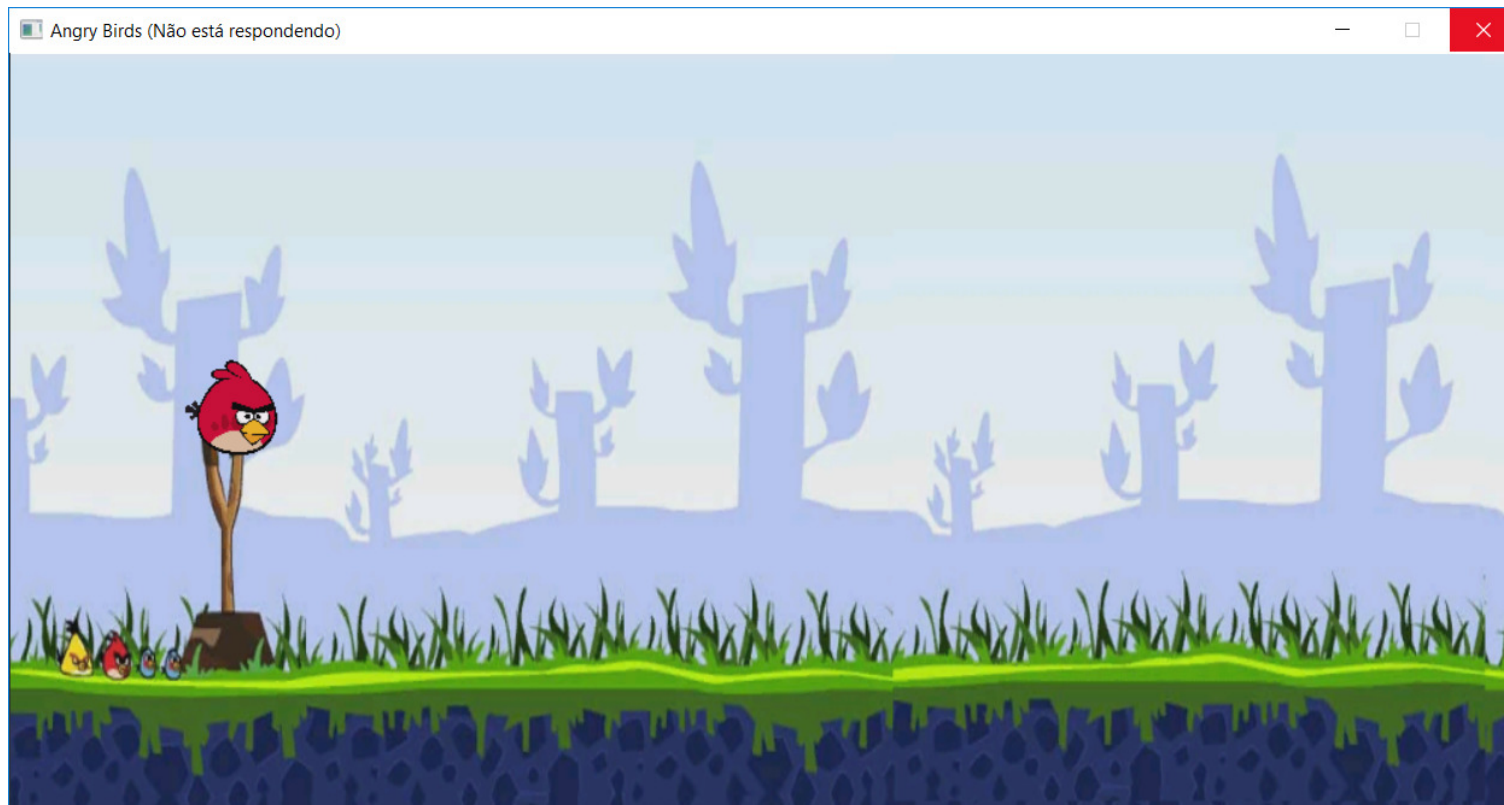
5. Manipulação de Imagens

5.1 Carregar um fundo

- `AssociaImagem()`
- `AbreImagem()`

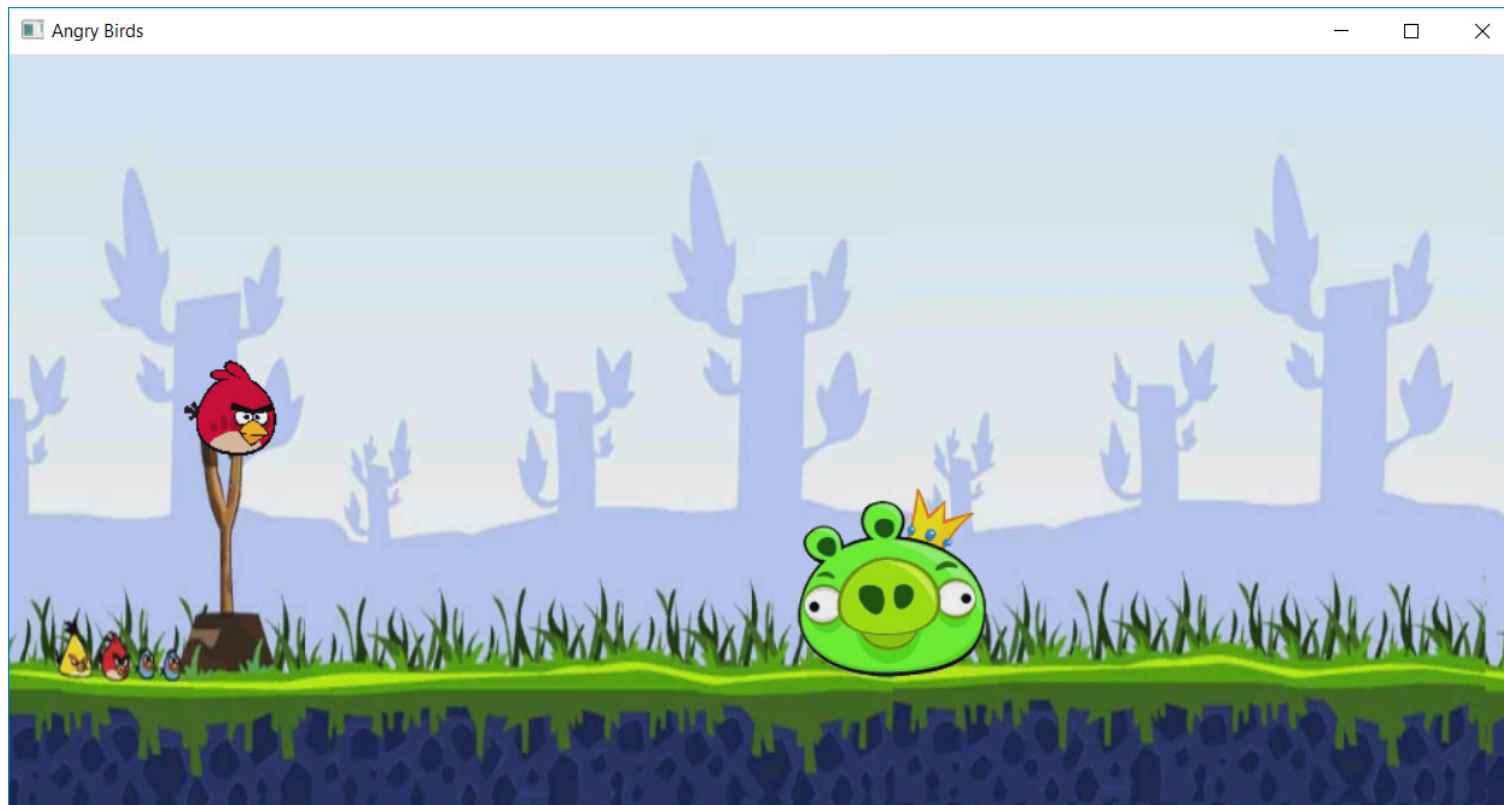
5. Manipulação de Imagens

5.2 Carregar um personagem-herói e posicioná-lo sobre uma estrutura elevada no canto esquerdo da janela.



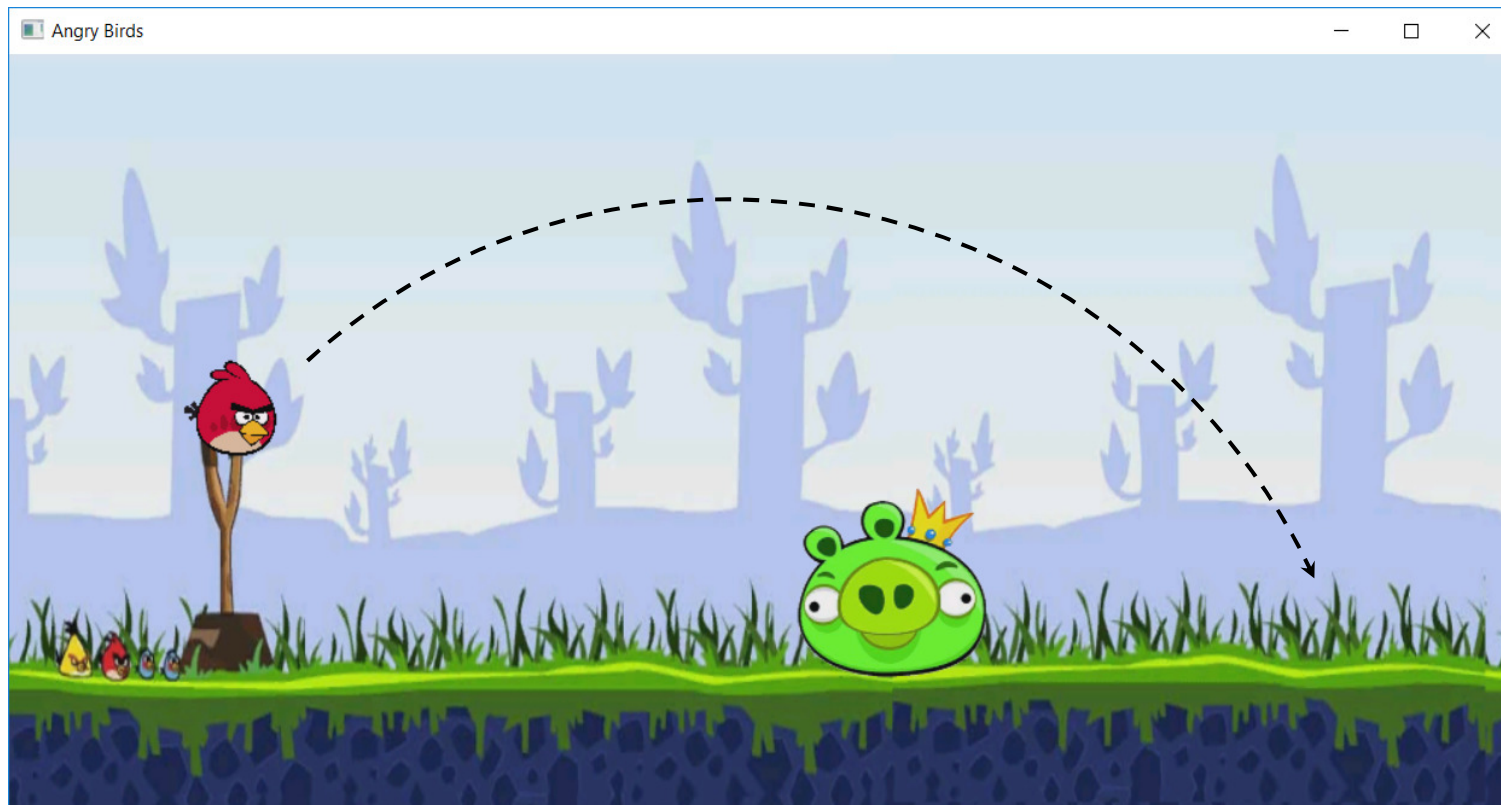
5. Manipulação de Imagens

5.3 Carregar um personagem-vilão e posicioná-lo em uma lugar aleatório sobre o chão.



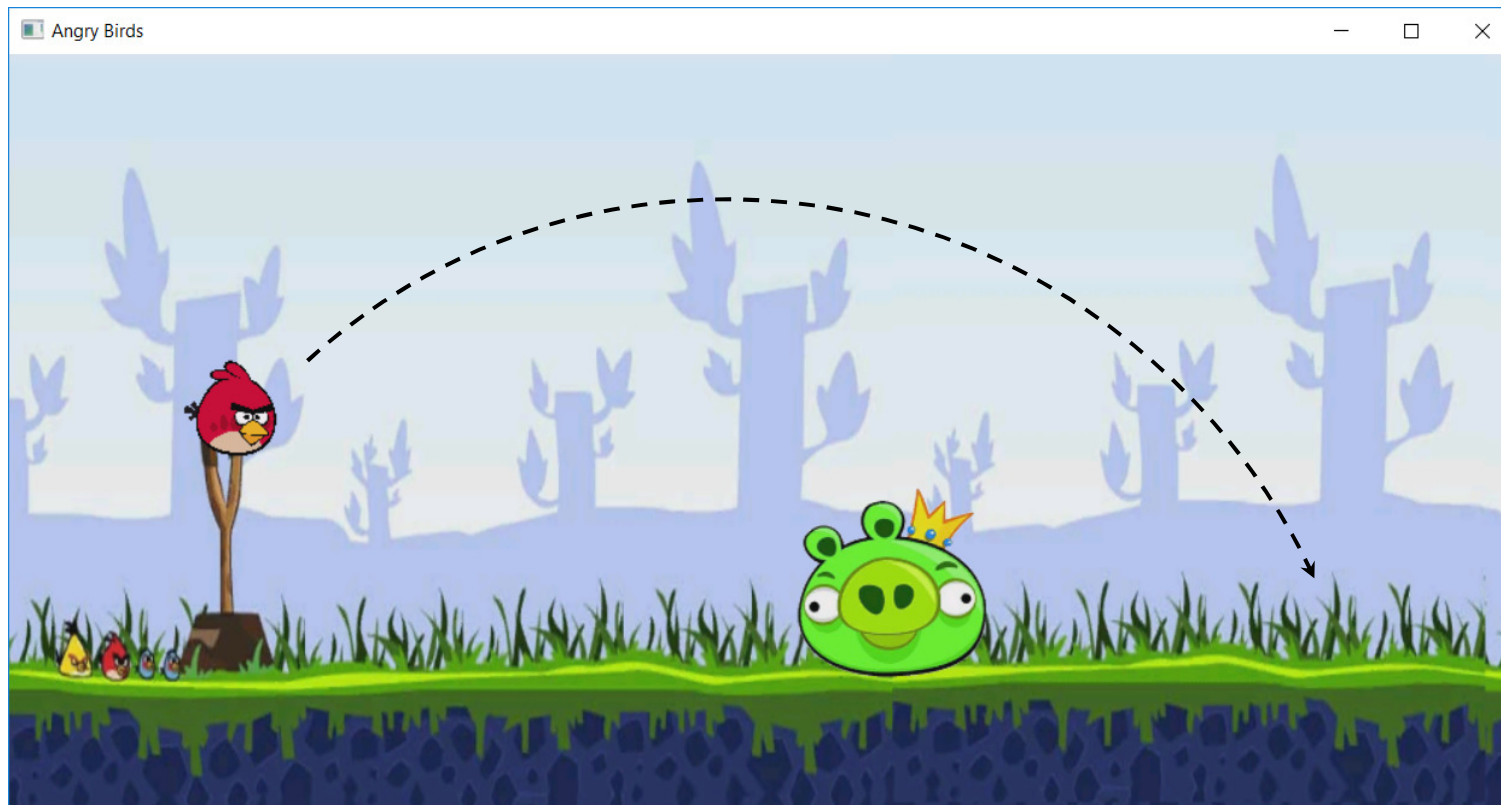
6. Movimentar o Personagem-herói

6.1 Aplicar uma trajetória balística (velocidade e direção iniciais definidas no código fonte).



6. Movimentar o Personagem-herói

6.2 Aplicar uma trajetória balística (solicitar ângulo e velocidade ao usuário).



7. Verificar Condição de Vitória

7.1 Verificar se há Colisão entre Vilão e Herói.

7.2 Fazer animação a partir da troca de imagens.



7. Verificar Condição de Vitória

7.1 Verificar se há Colisão entre Vilão e Herói.

7.2 Fazer animação a partir da troca de imagens.



7. Verificar Condição de Vitória

7.1 Verificar se há Colisão entre Vilão e Herói.

7.2 Fazer animação a partir da troca de imagens.

