

Universidade Federal de Santa Maria  
Disciplina: Computação Gráfica  
Primeiro Semestre de 2021  
Prof. Cesar Tadeu Pozzer  
Data: 06/07/2021

## Trabalho 3- Transformações Geométricas

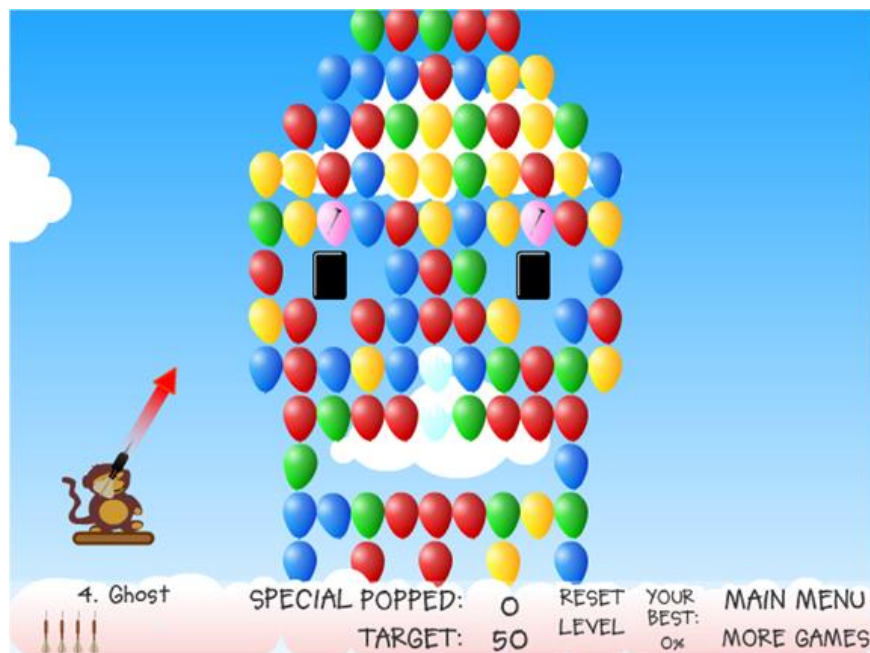
### Ferramentas:

Linguagem C++, utilizando a API Canvas2D (disponível no [site da disciplina](#)) e IDE Code::Blocks, compilando com MinGW (disponível na [versão 17.12 da IDE Code::Blocks](#)).  
**Não podem ser utilizadas bibliotecas auxiliares. Não pode ser usada a API OpenGL.**

### Descrição:

O jogo More Ballons consiste em disparar um dardo contra um conjunto de balões.  
[https://www.jogos360.com.br/more\\_bloons.html](https://www.jogos360.com.br/more_bloons.html).

O jogador controla um canhão e deve atirar projéteis. O jogador deve poder realizar a rotação do canhão, além de possibilitar ao usuário modificar a “força” com que o canhão dispara o projétil. A seguinte figura foi retirada do link acima e ilustra as definições acima citadas.



A orientação do projétil lançado deve ilustrar a trajetória que está seguindo, ou seja, não utilize um círculo como projétil, visto que não permite ilustrar a direção que o projétil está seguindo; utilize retângulos, flechas, figuras que dêem a impressão de direção. Os balões podem ser aproximados por círculos.

Deve-se trabalhar com vetores e suas propriedades. Procure imitar o funcionamento e formas de interação que ocorrem no jogo original. Para isso, pratique um pouco jogando o jogo.

A trajetória do projétil deve ser definida por uma curva de Bezier, que pode ser de grau 2. A curva deve ser gerada a partir da orientação e força do canhão.

**Critérios que serão avaliados:**

- Classes em C++ para definição de vetores e transformações geométricas
- Sistema do controle do canhão
- Sistema de movimentação do projétil. De preferência utilize o mouse para isso.
- Colisão do projétil com os alvos (círculos)

**Critérios básicos de interação:**

- Controle da direção do canhão
- Controle da força de disparo

**Critérios avançados de interação (Nota acima de 9.0):**

- Tipos diferentes de tiros (efeitos especiais, pesos diferentes, etc.) (+1,0)
- Tipos diferentes de alvos (alvos móveis, alvos que mudam trajetória do projétil, etc.) (+2,0)
- Inclusão de obstáculos, como ocorre no jogo (+1,0)
- Base do canhão móvel (+1,0)
- Criação de várias fases do jogo (+1,0)
- Concatenação de matrizes de transformação. (+2,0)
- Etc.

O trabalho deve apresentar uma **lista de instruções**, explicando de forma como o usuário deve interagir com o programa. Enumere no início do código fonte (arquivo main.cpp) os quesitos que foram implementados.

O trabalho deve ser individual. Dúvidas entrem em contato com o monitor ou comigo. A avaliação também poderá ser oral (a critério do professor).

## **Entrega:**

### **Formato de entrega:**

- O arquivo deve ter o nome do aluno (ou uma abreviação nome+sobrenome, ou seja, algo que identifique bem o aluno). O arquivo deve ser enviado pelo Google Classroom, até a data limite.
- Dentro deste arquivo deve haver um diretório com o mesmo nome do arquivo e, dentro deste diretório, os arquivos do trabalho.
- Deve-se enviar somente: código fonte (.cpp, .h, .hpp) e o projeto (.cbp). Não devem ser enviadas libs, executáveis, DLLs. Deve-se enviar a Canvas2D.

Antes do envio, teste se o projeto contido no seu .rar funciona em qualquer diretório que ele seja colocado.

### **Critérios de avaliação:**

- Documentação: descrever no cabeçalho de cada arquivo a ideia geral do código e comentar o que cada método e classe faz.
- Clean code: estrutura do código e nomeação de métodos, classes e variáveis devem ser fáceis de ler e entender. Procurar manter o código o mais simples e organizado possível.
- README: incluir um arquivo "README.txt" contendo informações sobre quais funcionalidades foram implementadas (requisitos e extras) e instruções de uso do programa caso o aluno julgue necessário ou caso tenha sido implementado uma funcionalidade extra que exija explicação.
- Pontualidade: Trabalhos não entregues na data não serão avaliados e receberão nota zero.
- Funcionalidade: o programa deve satisfazer todos os requisitos. Programas que não compilarem ou que não atenderem nenhum requisito receberão nota 0 (zero).