

## **Projeto Final**

### **Space Shooter utilizando a Biblioteca de Threads Desenvolvida**

#### **1. Descrição**

O projeto deve implementar o Space Shooter utilizando a biblioteca de Threads no espaço de usuário desenvolvida durante o semestre. O link abaixo demonstra as funcionalidades do jogo:

<https://www.youtube.com/watch?v=tDHvE9Aj8VI>

Para este trabalho, deve-se considerar um subconjunto das funcionalidades do jogo original, conforme as seguintes definições:

- **A espaçonave do usuário deve ser uma Thread;**
- **A criação das espaçonaves inimigas roxas em formato de “<”, que só se movimentam para frente, deve ser tratada em outra Thread;**
- **O desenho da tela deve ser feito por uma classe Window que será uma Thread (possivelmente de maior prioridade). A classe Window atualizará a tela usando o Timer da biblioteca gráfica a cada FPS ou quando uma tecla for apertada (tratamento de eventos da biblioteca);**
- **O tratamento da entrada do teclado deve ser realizado por outra Thread;**
- **Geração das minas deve ser feita por uma Thread a cada 30 segundos (note que elas explodem e jogam tiros para todos os lados se não forem atingidas em 5 segundos);**
- **Criação da Thread do “Boss” após 1 minuto de jogo;**
- **Controle de colisões e das 3 vidas através de uma Thread;**

Os alunos têm a liberdade para criar outras Threads como acharem necessário.

As teclas de entrada são as seguintes:

- **Teclas para cima, para baixo, lado esquerdo e lado direito para controlar a espaçonave do usuário.**
- **Tecla de espaço para disparar um tiro comum (que some ao matar o primeiro inimigo) e ‘1’ para disparar o míssil (que mata todos que colidir até o final da tela).**
- **ESC para sair do jogo.**

**Bônus de 1 ponto: tratamento de algum outro tipo de espaçonave (branca, laranja ou amarela) em outra Thread.**

A biblioteca gráfica utilizada será a Allegro (<https://liballeg.org/>). Para instalar a biblioteca no Ubuntu digite:

```
sudo add-apt-repository ppa:allegro/5.2
sudo apt-get install liballegro*5.2 liballegro*5-dev
```

```
sudo apt-get install liballegro-ttf5-dev
```

A documentação da biblioteca encontra-se em <https://github.com/liballeg/allegro/wiki/wiki/>

Cada grupo terá a liberdade para desenvolver o software seguindo o paradigma de orientação a objetos do C++ obrigatoriamente. Isso significa que o uso de programação estruturada e elementos da linguagem C terão descontos na nota (veja avaliação abaixo).

O Timer da biblioteca é abstraído na classe Timer (Timer.cc e Timer.h).

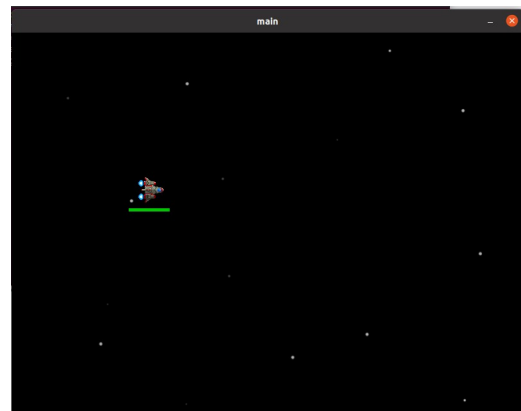
## 2. Arquivos Disponibilizados

Foram disponibilizados no moodle arquivos para criação de uma janela e desenho do background e da espaçonave do usuário na tela utilizando a biblioteca Allegro. Dentro do diretório “resources” são disponibilizados todos os arquivos para desenhar na tela.

Os desenhos foram retirados de <https://github.com/cbarnson/2d-space-shooter>. Esse repositório pode servir para retirar ideias (**não copiar código**) de como implementar.

O exemplo disponibilizado também já mostra o tratamento das teclas pela biblioteca:

```
./main
```



## 3. Formato de Entrega

Todos os arquivos utilizados na implementação do trabalho devem ser entregues em um único arquivo .zip ou .tar.gz na atividade do moodle. Deve ser anexado um arquivo Makefile para compilar o código e um **relatório descrevendo o projeto do software** (diagramas UML – casos de uso, classes e sequência da Thread da espaçonave pelo menos).

## 4. Data de Entrega

Data e horário da entrega estipulados na tarefa do moodle.

## 5. Avaliação

A avaliação se dará em 3 fases

1. Avaliação de compilação: compilar o código enviado. Caso haja erros de compilação, a nota do trabalho será automaticamente zerada.
2. Avaliação de execução: para validar que a solução executa corretamente sem falhas de segmentação. Caso haja falhas de segmentação, a nota é zerada. Será também avaliado o uso de variáveis globais (-5 pontos) e vazamentos de memória (-20%). **Variáveis estáticas de classes não são consideradas globais.**
3. Avaliação da organização do código: busca-se nesta fase avaliar a organização do código orientado a objetos. Deve-se usar classes e objetos e não estilo de programação baseado em procedimentos (como na linguagem C). Uso de programação estruturada (-30%). Uso da linguagem C ao invés de linguagem C++ (-30%)

Este trabalho precisará ser apresentado aos professores nos dias 13 e 16/12/2022 em sala de aula (entrega dia 11/12/2022). Plágio não será tolerado em nenhuma hipótese ao longo dos trabalhos, acarretando em nota 0 a todos os envolvidos. Cada grupo terá entre 5 e 8 minutos para apresentar, devendo apresentar primeiramente a estrutura do software projetado para o jogo (1 figura) e em seguida apresentar o jogo funcionando.

Haverá votação para a escolha do melhor jogo, sendo este premiado com 1 ponto na nota final do projeto.

Os pesos da avaliação serão os seguintes:

- 30% para o relatório
- 10% para a apresentação
- 60% para a avaliação funcional e organização do código (seguindo as regras acima)