

# Estruturas de Dados

## Trabalho 2

Prof. Bruno Azevedo

Instituto Federal de São Paulo



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SÃO PAULO  
Campus Catanduva

# Trabalho 2

- Implementar um RPG em C++ de acordo com as seguintes instruções:
- O jogo será um RPG onde um herói, controlado pelo usuário, explorará um ou mais labirintos encontrando tesouros e inimigos. Idealmente, o jogo deve reutilizar código de T1.
- Os arquétipos criados para T1 poderão ser utilizados para serem as classes selecionadas pelo usuário ou dos inimigos no labirinto.
- Terá as mesmas mecânicas de ataque especial e sangramento, como implementadas em T1.
- O usuário irá controlar o personagem com o teclado e/ou mouse enquanto explora o labirinto e luta contra os inimigos.
- O combate pode possuir a mesma implementação de T1 ou pode permitir que o usuário tome decisões durante o combate.
- Na tela inicial o usuário deve escolher o nome e a classe do personagem.
- O personagem deve possuir um inventário, onde guardará itens e tesouros encontrados no labirinto.
- Os itens devem prover bônus para o personagem durante os combates. Por exemplo, um escudo pode prover aumento da defesa.
- O labirinto deve ter um fim onde o personagem lutará com um *boss* para finalizar o jogo.
- Ao fim, o jogo deve apresentar uma pontuação de acordo com os tesouros acumulados durante a exploração do labirinto.

# Trabalho 2

- Trabalho em grupos de até 3 integrantes.
- Seu código obrigatoriamente deve ser feito em C++ e o conhecimento obtido até o momento. **Utilizem as estruturas de dados ensinadas em aula.**
- Reutilizem código do trabalho 1. Não é obrigatório, mas facilitará o esforço de desenvolvimento.
- Em T1 vocês tiveram que pensar em como modelar o torneio (qual estrutura de dados utilizar, como utilizar), como modelar a lista de inimigos derrotados, como modelar os arquétipos, etc. Neste trabalho, pensem em como modelarão o jogo utilizando as estruturas de dados aprendidas.
- Algumas sugestões:
  - Filas podem ser utilizadas em sistemas de turnos.
  - Filas de prioridade permitem o gerenciamento de eventos de diferentes prioridades.
  - Listas ligadas podem ser utilizadas para gerenciar inventário, inimigos, etc.
  - Pilhas permitem o gerenciamento de bônus de equipamentos ou de bônus temporários causados por itens que expiram como poções, ou seja, utilize uma pilha de bônus.
  - Pilhas também podem ser utilizadas para armazenar ações e permitir desfazer ações efetuadas.
- Estruturas de dados permitem armazenar, organizar e gerenciar conjuntos dinâmicos de dados. **Utilizem-as para este fim, considerando os dados existentes em seu jogo.**
- Data de entrega: a ser definida.
- Idealmente, haverá uma ou duas aulas para os grupos apresentarem os trabalhos para o professor. Não será seminário.

# Três Sugestões Ajudar no Desenvolvimento da Parte Gráfica

- O jogo deve possuir um componente gráfico. Ou seja, deverá exibir o personagem se movimentando e interagindo com o ambiente do jogo.
- Para este fim, passarei três sugestões para ajudar neste esforço.
  - SFML.
  - ncurses.
  - libtcod.

# Simple and Fast Multimedia Library (SFML)

- <https://www.sfml-dev.org/>.
- Tutoriais: <https://www.sfml-dev.org/tutorials/2.6/>.
- GitHub: <https://github.com/SFML/SFML>.
- Livro: SFML Game Development By Example.

# New Curses (ncurses)

- `https://invisible-island.net/ncurses/`.

# libtcod

- <https://github.com/libtcod/libtcod>.
- <https://libtcod.readthedocs.io/en/latest/>.
- Documentação mais completa, da versão 1.6.4:  
<https://libtcod.github.io/docs/index2.html?c=true&cpp=true&cs=false&py=false&lua=false>.