



UNIVERSIDADE FEDERAL DO CEARÁ – UFC SOBRAL  
TÉCNICAS DE PROGRAMAÇÃO – PROF. FISCHER

## Prova 2:

### O Regaste dos Alunos na Ilha de Java - Versão 2

A Prova 2 consiste na **implementação** da interface gráfica do jogo, como visto em sala de aula.

Front-end da aplicação deverá ser feito tendo como base exemplo da Figura 1. Obs: A Figura 1 apresenta apenas um exemplo do jogo, que poderá ser alterado se você achar melhor, porém as funcionalidades e mecânicas do jogo deverão ser mantidas.

Inicialmente, o nome do jogador deverá ser requerido pelo jogo. Depois do nome inserido o jogador poderá iniciar o jogo por meio do botão “**Jogar**”. O tabuleiro do jogo deverá ser demonstrado, quando o botão “**Jogar**” for clicado. Ainda, deverá ser exibido o menu de opções de robôs, o quadro de pontuação com o número de alunos e bugs encontrados pelos robôs ao longo do jogo e os botões de “**Verificar**”, “**Próxima Jogada**” e “**Sair do Jogo**”.



Figura 1: tela principal do jogo

Como demonstrado em sala de aula, o jogo consiste na escolha, por meio do jogador, em qual célula do tabuleiro cada robô deverá ser posicionado. Depois que o jogador indicar a célula para cada robô, o botão “**Verificar**” deverá ser clicado pelo jogador.

Devidos tratamentos para o jogo funcionar corretamente deverão ser implementados. Por exemplo, se o jogador clicar em "**Verificar**" sem ter indicado para qual célula cada robô deverá ficar, uma mensagem deverá ser exibida para que o jogador posicione os robôs antes do jogo verificar as posições.

O jogo deverá computar para cada posição escolhida se existe um bug ou aluno. Assim, se na posição escolhida for encontrado um bug o contador de bug deverá ser acrescentado (número ao lado da imagem do bug) se por acaso um aluno for encontrado um contador de aluno deverá ser acrescentado.

Lembrando que as células que ainda não foram abertas não deverão ficar visíveis, se existe um bug, aluno ou nada dentro da célula.

A cada aluno resgatado o jogador ganha **10 pontos**. Para resgatar o aluno o robô tem que estar na mesma célula que o aluno se encontra. Porém, se o robô for movido para uma célula em que tenha um bug, o jogador perde **15 pontos**.

Quando o jogador clicar no botão "**Próxima Jogada**" novamente os robôs poderão ser posicionados nas células ainda não visitadas.

Para cada célula **visitada** que tenha um aluno deverá ficar visível no tabuleiro o aluno dentro na célula. Da mesma forma, para a célula **visitada** que tenha um bug, o bug deverá ficar visível na célula. Para a célula visitada, que não tenha robô ou aluno, deverá ser indicada com algum destaque. Por exemplo, a célula visitada poderá ficar com *background* diferente das demais. Ainda, a célula visitada não mais poderá ser escolhida pelo jogador.

Quando o usuário sair do jogo por meio do botão "**Sair do Jogo**" uma tela contendo um **relatório do jogo** deverá ser apresentada, com as seguintes informações:

- 1) Nome do jogador
- 2) Quantidade de rodadas ocorridas no jogo (obs: cada vez que o jogador clicar próxima jogada é contado uma unidade para essa métrica)
- 3) Pontuação do jogador
- 4) Quantidades de células vazias clicadas
- 5) Quantidade de alunos resgatados
- 6) Quantidade de bugs encontrados
- 7) Pontos individuais de cada robô

Todos os dados do **relatório do jogo** deverão ser armazenados em arquivo de texto. Assim, se o usuário clicar no botão "**Relatório do Jogo**" todas informações de todos os jogadores (que já jogaram em alguma momento) deverão ser exibidas em uma tabela da seguinte forma, como o exemplo da Tabela 1.

Tabela 1: exemplo de relatório do jogo

Jogador	Quant. Rodadas	células vazias	Pontuação	Alunos resgatados	Bugs encontrados	Andador	Cavalo	Rei
Cauã	20	10	70	10	2	100	-15	-15
Geovana	30	15	120	15	1	50	35	50

Desta forma, quando o jogo iniciar um arquivo de texto com os dados dos jogadores anteriores deverão ser carregados para a memória do computador.

Obs: o arquivo com os dados do jogadores somente cresce, mas, nunca diminui.

Para a Tabela 1 o arquivo com os dados do jogadores está organizado da seguinte forma:

**Cauã;20;10;70;10;2;100;-15;-15;**  
**Geovana;30;15;120;15;1;50;35;50;**

#### 1. Requisitos para a entrega da Prova 2

- Data: 25/06/2023
- Pontuação: 25 pontos
- Trabalho individual
- Poste seu código no GitHub
- Faça um vídeo de explicação e poste no YouTube como não listado
- Coloque no classroom o link do seu projeto no GitHub e link do vídeo de explicação do seu trabalho
- Você deverá gravar um vídeo entre 13 a 15 minutos explicando o seu trabalho. As apresentações fora desse intervalo (13 a 15 minutos) não serão pontuadas
- Exemplos de ferramentas para gravação: ActivePresenter, câmera do celular e etc...
- No início do vídeo você deverá se apresentar (nome, matrícula e período em que você está cursando) “filmando seu rosto”
- Demais partes do vídeo podem ser apenas a apresentação da sua tela na qual você estará explicando o seu código

Observação importante: Trabalhos que não serão pontuados:

1. Trabalho que não tenha o vídeo de apresentação
2. Trabalhos duplicados para mais de um aluno
3. Trabalhos que não sigam as instruções do jogo acima descritas
4. Trabalhos que não sigam o paradigma da Orientados a Objetos

Bom trabalho para todos!