



Universidade Federal Rural de Pernambuco – UFRPE

Curso: Bacharelado em Sistemas de Informação – BSI

Disciplina: Introdução à Programação – Turma SI2 - Prof: Leopoldo Teixeira

### Segundo Projeto – 2014-1 – Valor: 25% (25 pts) da 2ª V.A.

#### Instruções:

- O projeto deve ser feito em grupos de até 2 alunos.
- Data de Entrega: 03/08/2014 até às 23h59m por email.
- Apresentação pelo grupo: dia 01/08/2014 no horário da aula.
- Até a data de entrega, fazer o upload do projeto em uma conta no GitHub ([www.github.com](http://www.github.com)) e enviar o link para o email [leo@leopoldomt.com](mailto:leo@leopoldomt.com) com o assunto: [Projeto2-IP-2014-1]. Informar os nomes dos componentes do grupo no conteúdo do email.
- As equipes podem (e devem!) tirar dúvidas com o professor antes da entrega do projeto, apresentando as dificuldades encontradas para discutirmos possíveis ideias e soluções;
- Apresentar o código ao professor, em sala, no dia da aula
- Condições para receber nota 0 (zero):
  1. Entrega fora do prazo estabelecido;
  2. Código com erros de sintaxe;
  3. Programa incompleto;
  4. Código com alta similaridade com algum de outro(s) grupo(s);
  5. Não saber explicar o código.

A equipe irá estruturar o jogo inspirado em “Batalha Naval” desenvolvido como primeiro projeto utilizando os conceitos vistos na segunda parte da disciplina. As especificações do projeto estão abaixo.

```
# criar uma funcao que pergunta ao usuario se deseja configurar o jogo informando
# atencao para o tipo de retorno que esta funcao deve ter
if ConfigurarAleatoriamente():
    # define as variaveis do jogo com base no arquivo de configuracao (config.txt)
    # os valores devem respeitar as mesmas regras definidas no primeiro projeto
    # neste caso a posicao dos navios no grid sera gerada aleatoriamente a partir
    # das variaveis, assim como no primeiro projeto
else:
    # define as variaveis do jogo a partir de um arquivo informado pelo usuario,
    # no formato disponivel no site (grid.txt)
    # atente para o formato do arquivo, a quantidade de erros esta na 1a linha
    # em seguida, após uma linha em branco, o grid com as posições entre espaços
    # navios devem estar representados com a letra N
    # posições do mar devem estar representados com "."
    # a matriz deve ser montada a partir do arquivo, assim como a variavel
    # que armazena a quantidade de navios sera definida com base no total de
    # navios definidos no grid

# em seguida defina funcoes que implementem a funcionalidade do jogo
1. exibir uma matriz para o usuário no formato definido no primeiro projeto
2. atirar em uma coordenada informada pelo usuário. Atendem para a validação,
devemos verificar se a coordenada existe, se já atirou nela, se corresponde a um
navio ou água...
3. verificar se a quantidade total de navios já foi atingida
4. verificar se a quantidade limite de erros já foi atingida
5. também deveremos manter a pontuação do jogador. Ao acertar um navio, a
pontuação aumenta +10, ao errar a pontuação diminui -1
6. ao final do jogo, a pontuação do jogador deverá ser salva no fim do arquivo
scores.txt, com o nome, que será informado pelo jogador
7. antes do programa ser encerrado, o ranking de pontuações deve ser exibido em
ordem decrescente, isto é, do maior para o menor
```

A equipe é livre para fazer versões mais elaboradas do jogo, podendo ser bonificada com ponto extra de acordo com o esforço adicional de implementação destas funcionalidades.