

## Programação para Internet Rica

### Trabalho B

Como visto em sala, as Aplicações Ricas para Internet são aquelas que propiciam interação e possuem características similares às aquelas de programas desktop, usando a plataforma Web.

Aplicações mais complexas exigiriam um grande trabalho conjunto de desenvolvimento cliente-servidor, que foge ao escopo do curso, além de um excessivo trabalho em número de horas. Sendo assim, como um jogo é desenvolvido ao longo do semestre, é proposto o **desenvolvimento do lado cliente de um jogo**.

**Objetivo:** Desenvolver a parte visual do lado cliente de um jogo.

**Tecnologias:** XHTML Strict, CSS Level 2 ou 3, JavaScript, DHTML, DOM, JQuery, AJAX.

#### Orientação:

- 1) Pegue o resultado do trabalho A e, caso não esteja pronto o XHTML e CSS, finalize.
- 2) Identifique quais são as informações **mínimas necessárias** que o navegador deve fornecer nas seguintes situações:
  - a. Início de Jogo; (ex.: no jogo da velha, precisamos indicar se o usuário escolheu **x** ou **o**)
  - b. Um dos jogadores faz um movimento. (ex.: no jogo da velha, precisamos indicar a célula clicada)
- 3) Proponha comandos para enviar tais informações ao servidor. Exemplo: no jogo da velha usamos:
  - a. start=x , start=o
  - b. cel=c0 , (...), cel=c8
- 4) Identifique quais são as informações **mínimas necessárias** que o servidor precisa fornecer como resposta aos comandos enviados pelo navegador, para que o estado atual do jogo possa ser desenhado no navegador (tabuleiro). Exemplo: no jogo da velha usamos uma string de 10 caracteres: os 9 primeiros indicam as células c0 a c8 e a décima indica o estado do do jogo. Para os 9 primeiros caracteres, os valores podem ser:
  - a. **[espaço]**: célula vazia.
  - b. **x, o**: célula ocupada com um X ou O.
  - c. **X, O**: célula ocupada com um X ou O e que faz parte da trinca da vitória.O 10º caractere indica:
  - a. **[espaço]**: jogo ainda em andamento.
  - b. **v**: jogo finalizado, jogador venceu.
  - c. **p**: jogo finalizado, jogador perdeu.
  - d. **e**: jogo finalizado, deu empate.

- 5) Construa a função **pintaTabuleiro** que use as informações do estado do jogo (vindas do servidor), conforme vocês definiram seguindo as regras do item 4, para preencher o tabuleiro do XHTML/CSS.
- 6) Construa as funções relativas à jogada do usuário (ex.: **click** no jogo da velha), de acordo com as regras do jogo escolhido e conforme as informações definidas por vocês no item 3. Lembre-se que se o movimento do jogador exige 2 cliques (“peça a mover” e “destino da peça”, por exemplo, como no jogo de Xadrez) você precisará de duas rotinas de clique, que devem ser usadas alternadamente. Use uma variável global do JavaScript (similar à variável **jogador** do jogo da velha) para controlar se é o primeiro ou o segundo clique do jogador. **Não é necessário verificar se o movimento é válido ou não.**
- 7) Ainda que a(s) função(ões) **click** tenha(m) algum efeito visual direto, ela(s) deve(m) ser implementada(s) para comunicação com servidor via **AJAX**.
- 8) Crie um campo de **debug** usando a criação dinâmica de elementos DOM, para permitir a simulação do recebimento de informações por parte do servidor.
- 9) Crie um placar para o jogo.
- 10) Faça as melhorias que julgar necessárias ao jogo, usando os recursos do JQuery.

**Entrega:** a entrega deve ser feita no formato **ZIP (não use RAR)** pelo WebAula (no SIA, entre no Campus Virtual, Minhas Disciplinas Presenciais, selecione a disciplina Programação para Internet Rica e depois selecione o ícone Trabalhos, onde deverá haver uma opção para entrega do Trabalho B. A data de entrega está definida na apresentação PowerPoint da primeira aula e, como já devidamente avisado, atrasos implicarão em descontos na nota.

Dentro do arquivo ZIP devem constar todos os arquivos necessários para executar o seu jogo (a parte Cliente, apenas) e um arquivo DOC/DOCX descrevendo os comandos e respostas do servidor, conforme solicitado nos itens 2 a 4.

**A avaliação final será dada por:**

- a) Qualidade visual e efeitos (0,5 pontos).
- b) Limpeza e correção do código, incluindo o uso de boas práticas (0,5 pontos).
- c) Implementação de todos os recursos dos itens 1 a 10 (0,5 pontos).
- d) Documentação completa e correta (0,5 pontos).