

## **Implementação de um jogo multi-jogador online (MMOG) usando o Phaser: Exclusão Mútua**

O Trabalho Prático 2 a ser realizado na disciplina de SD consiste no projeto e implementação de um **jogo multi-jogador online**. O jogo deverá rodar em dispositivos móveis, para isto você irá programa-lo usando o framework Phaser (<http://phaser.io/>). Este documento especifica os requisitos do serviço.

### **1. Cronograma e Valor**

O trabalho vale 10 pontos no total. Ele deverá ser entregue até o dia 15 de outubro às 23:55h.

### **2. Regras**

- O trabalho poderá ser realizado em equipes de até 4 alunos. Entretanto, a experiência indica que a melhor formação é em trio.
- A implementação deverá ser realizada usando o Phaser.
- A organização do relatório será considerada para fins de avaliação.
- A tolerância para entrega com atraso é de 4 dias, com perda de 20% da nota inicial por dia.
- Os trabalhos somente serão recebidos via Moodle.
- No horário da aula seguinte caso, será feita a apresentação do trabalho por parte das equipes, lembrando que a nota é coletiva.

### **3. Código de honra**

- **Já que não faz sentido copiar código de outro aluno, pois a idéia é trabalhar para aprender, trabalhos total ou parcialmente iguais receberão avaliação nula.** Para isto serão executados testes automáticos sobre o código entregue.
  - Trabalhos total ou parcialmente iguais a projetos apresentados por outros alunos em semestres anteriores receberão avaliação nula (exceto se for o trabalho do próprio aluno).
4. O monitor poderá realizar arguição com os alunos a respeito do trabalho elaborado.

### **Problema**

Jogos multi-jogador online em Sistemas Distribuídos permitem que vários jogadores participem simultaneamente de uma mesma partida (Wikipedia). O jogo pode ser cooperativo ou competitivo. Nos jogos *Cooperativos* ambos jogadores trabalham em equipe para atingir um objetivo em comum. Já nos *Competitivos* os jogadores trabalham um contra o outro e quem alcançar o objetivo primeiro, vence.

A arquitetura de jogos massivos multi-jogador pode variar de um tipo de jogo para

outro. Por exemplo, jogos de estratégia podem ser implementados baseados em uma arquitetura híbrida combinando as arquiteturas P2P e Cliente-Servidor. Vários jogadores são agrupados em uma sala usando uma arquitetura P2P, os dados da sala são registrados em um servidor centralizado (ex. melhores scores dos jogadores).

Nesse trabalho você deverá:

- Começar a implementar um jogo online que permita jogar online a usuários em dispositivos móveis. O número máximo de usuários do jogo é, em princípio, ilimitado. Jogos de 2 ou 3 usuários não serão aceitos;
- Nessa fase do jogo você irá implementar um **serviço de locks** que permita garantir, por exemplo, que 2 personagens não ocupam a mesma casa ao mesmo tempo;
- O jogo não precisa ficar bonitinho (mas seria melhor se ficasse ☺). O importante no seu trabalho é aplicar conhecimentos de sistemas distribuídos;
- Este programa será usado nos trabalhos seguintes. Projete ele com carinho.

## 5. O framework Phaser

O framework Phaser disponibiliza um tutorial rápido que dá uma passada pelas principais funcionalidades do framework, desde como trabalhar com *sprites* até a questão da movimentação (gravidade, elasticidade de objetos, etc.).

Arquivos do tutorial:

[https://github.com/photonstorm/phaser/raw/master/resources/tutorials/02%20Making%20your%20first%20game/phaser\\_tutorial\\_02.zip](https://github.com/photonstorm/phaser/raw/master/resources/tutorials/02%20Making%20your%20first%20game/phaser_tutorial_02.zip)

Tutorial: <http://phaser.io/tutorials/making-your-first-phaser-game/index>

Quanto à IDE, é possível utilizar qualquer editor de texto, visto que são arquivos .html, .js, etc.. Contudo, eles disponibilizam um editor online. É bem simplificado e dá para visualizar o projeto na hora. Além de possuir *autocomplete*, o que vai ajudar bastante.

Editor: <http://phaser.io/sandbox>

## 6. O que entregar

Nesse TP deverão ser entregues via Moodle:

- Código fonte.
- Relatório contendo:
  - Descrição breve do jogo e suas regras.
  - Projeto do aplicativo distribuído (algoritmo usado), e como compilar e executar. Não deve ser incluída a listagem do código fonte no relatório exceto trechos de código que a equipe considere relevantes para a explicação. O foco

do relatório está nos aspectos distribuídos do trabalho e não na parte de Engenharia de Software.

- Descrição do seu algoritmo de Exclusão Mútua e por que ele está correto (quais as propriedades que satisfaz e por quê).
- Seu serviço de locks é centralizado ou distribuído? Explique o motivo pelo qual você escolheu essa arquitetura para o seu serviço.

## **7. Avaliação (10 pontos)**

- 40% execução correta do programa
- 40% Relatório: projeto e desempenho
- 20% Qualidade do texto e conclusões da equipe

**Trabalhos que implementam algo diferente do pedido não serão avaliados**

**Dúvidas? Consulte o seu monitor ANTES da data de entrega!**

**Bom trabalho!**