



## **1. Introdução**

O objetivo deste trabalho é adaptar um jogo de War já existente (trabalho base) de modo que ele possa ser jogado por até 6 jogadores conectados em uma rede local de computadores.

## **2. Modelo de Distribuição e Comunicação**

O jogo deve ser organizado segundo uma arquitetura cliente servidor com os seguintes papéis:

1. Cada jogador realizará suas jogadas em um computador em que houver um programa cliente instalado;
2. O programa do cliente é responsável pela interface gráfica e pela implementação de todas as regras do jogo;
3. Após a realização de uma jogada o programa cliente deve reunir os dados que representam o estado de uma partida, transformá-los em uma cadeia de caracteres e enviá-los para o servidor;
4. O servidor é um mero mediador. Sua função é receber o estado do jogo de um cliente e repassá-los, imediatamente, para os demais jogadores;
5. Nenhuma regra do jogo será implementada o garantida pelo servidor. Entretanto, ele é o responsável por determinar, e informar a todos, quem é o jogador da vez;
6. Após receber do servidor o estado de uma partida, o cliente deverá atualizar a interface gráfica de modo a refletir o novo estado do jogo.

O modelo de comunicação entre os clientes e o servidor é o mesmo usado no trabalho do chat para reuniões virtuais. Dessa forma, os programas devem se comunicar através de portas sockets, disponíveis no protocolo TCP e implementadas nas classes Java ServerSocket e Socket.

### **3.Estruturas de dados**

As coleções e outras estruturas de dados necessárias devem ser implementadas usando-se o framework de coleções de Java.

### **4.Organização do Código e Design Patterns**

O programa deve ser organizado em pacotes e empregar ao menos três design patterns: Façade, Singleton e Observer.

### **5.O Processo de Desenvolvimento**

O trabalho deve ser desenvolvido com a **linguagem Java** e a **última versão** da ferramenta **Eclipse**. Além disso, será levada em consideração a aplicação correta das técnicas de design e programação vistas durante o curso. Isso inclui a observação dos critérios de acoplamento e coesão, a organização do aplicativo em pacotes e o emprego dos design patterns citados na seção 4 deste documento.

### **6.Artefatos a Serem Entregues**

O código fonte do jogo só precisará ser entregue na última iteração. Isto é, o código fonte deverá ser enviado para o professor até as **23h59** do dia **19/06/2016**, data da entrega do trabalho.

Ao longo das próximas semanas, entretanto, farei o acompanhamento do desenvolvimento dos trabalhos. Para tal, cada grupo terá de me enviar, semanalmente, um arquivo MS Word contendo o seguinte:

- Descrição das atividades realizadas por cada membro do grupo durante a semana;

- Descrição das atividades que serão realizadas por cada membro do grupo na semana seguinte.

Esses relatórios devem começar a ser enviados a partir do dia **29/05/2016**. As entregas devem se repetir todos os domingos até a data de entrega do trabalho (**19/06/2016**).

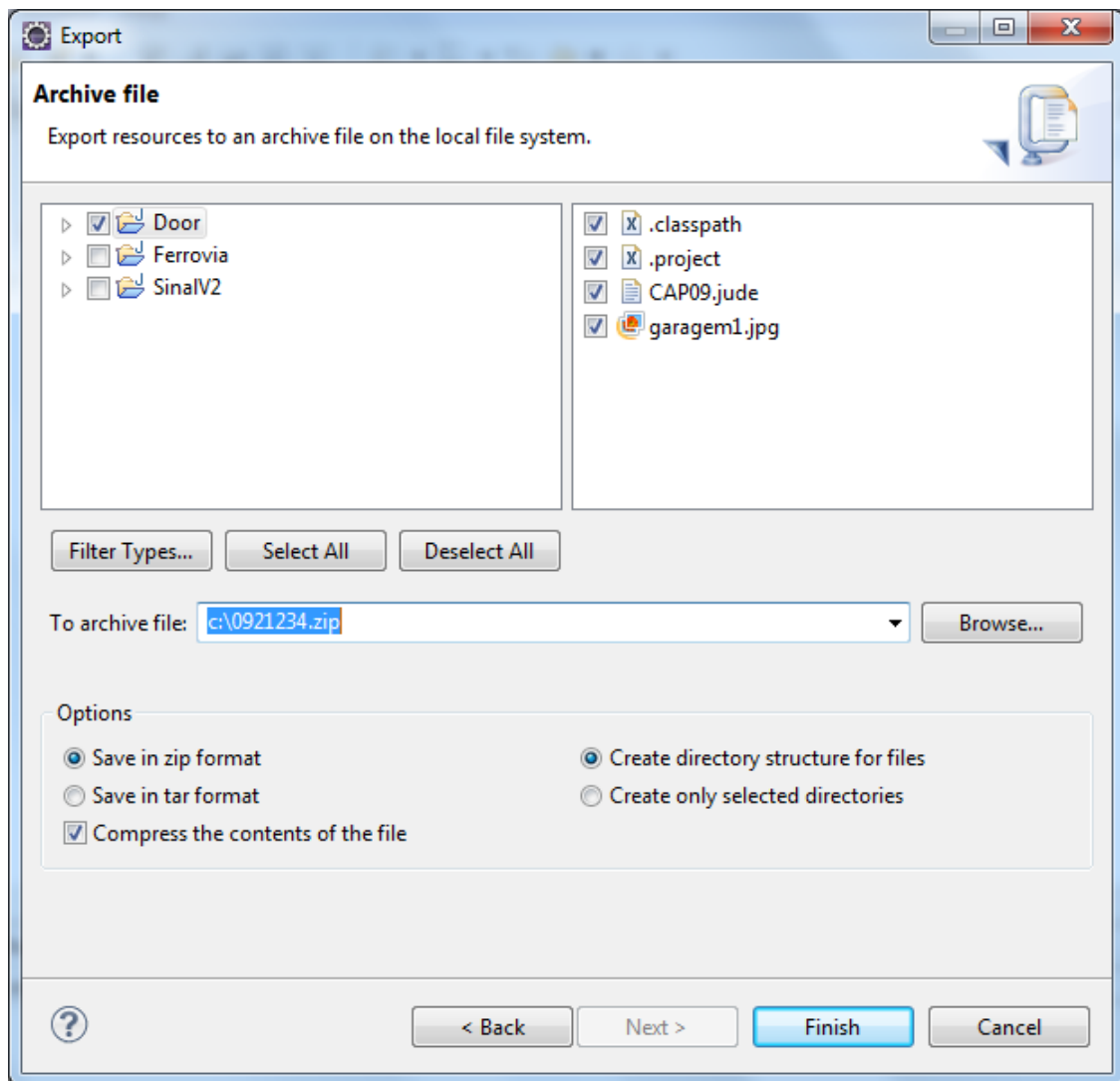
Juntamente com o 1º relatório, deve ser entregue um arquivo astah, ou jude, com um ou mais diagramas de classes e de pacotes resultantes da aplicação de engenharia reversa sobre o trabalho base. Esses diagramas irão ajudá-los a entender a arquitetura do trabalho base, tarefa fundamental para que se possa adaptá-lo para funcionar em ambiente distribuído.

A segunda versão do modelo de classes do jogo deve ser entregue junto com o 2º relatório (**05/06/2016**). Ele deve mostrar um esboço das classes que vocês irão inserir para viabilizar a comunicação dos clientes com o servidor. Um ou dois diagramas de sequência que ilustrem alguns detalhes das interações responsáveis por essa comunicação serão bem-vindos.

Ao término da última iteração (**19/06/2016**), juntamente com o relatório semanal de atividades, será feita a entrega do projeto Eclipse com a implementação do trabalho (códigos fonte Java).

O nome do projeto deve ser composto pela matrícula dos dois membros do grupo. Por exemplo, **1222345-1113456**.

O projeto deve ser entregue em um arquivo zip, como mostra a figura a seguir. O nome do arquivo deve ser composto pela matrícula dos dois membros do grupo. Por exemplo, **1222345-1113456.zip**.



## 7.Cronograma de Entregas

Todos os domingos, até **19/06/2016** – relatório das atividades realizadas na semana.

**29/05/2016** – arquivo astah, ou jude, com um ou mais diagramas de classes e de pacotes resultantes da aplicação de engenharia reversa sobre o trabalho base.

**05/06/2016** – arquivo astah, ou jude, contendo as classes do trabalho base acrescidas das classes que viabilizarão a comunicação dos clientes com o servidor.

**19/06/2016** – último relatório de atividades e o projeto Eclipse com a implementação do trabalho.

**20, 21 e 22/06/2016** – Apresentações dos trabalhos completos.

Todos os artefatos relativos às entregas deverão ser enviados para o e-mail do professor ([ivan.mathias.filho@gmail.com](mailto:ivan.mathias.filho@gmail.com)). **Atrasos nas entregas semanais acarretarão na perda de 1,0 ponto na nota final do trabalho por cada dia, ou fração do dia, de atraso, independentemente da qualidade da versão final.**

**Não haverá tolerância na entrega final (19/06/2016). Quem não enviar o trabalho completo até o dia 19/06/2016 receberá ZERO no G2.**

## **8. Apresentação**

**A não apresentação do trabalho em uma das datas de apresentação previamente definidas acarretará na atribuição da nota ZERO para o G2.**

O trabalho deverá ser apresentado nas máquinas do LabGrad. A distribuição dos grupos pelas datas irá obedecer à lista de chamada.

## **9. Regras Adicionais**

Os grupos serão de **no máximo dois alunos**. Trabalhos feitos por mais de dois alunos serão desconsiderados e **TODOS receberão grau ZERO**.

É proibido o compartilhamento de código entre os grupos. Caso fique evidente que um grupo copiou parte do código do trabalho de outro grupo, todos os alunos, de ambos os grupos, receberão grau **ZERO**. Posteriormente o caso será levado à Comissão Disciplinar do CTC.