



1. Introdução

O objetivo deste trabalho é adaptar um jogo de Ludo já existente (trabalho base) de modo que ele possa ser jogado por 4 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 ou garantida pelo servidor. Entretanto, ele será 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 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.Relatórios de Acompanhamento

Ao término de cada iteração cada participante terá de enviar um relatório descrevendo o andamento dos trabalhos. Os relatórios devem ser entregues por meio do Ambiente de Aprendizagem On-line da PUC-Rio. Eles devem ser individuais e seguir um modelo que será publicado posteriormente.

Cada componente do grupo deverá informar o que ele fez desde o término da última iteração e o que ele planeja fazer até o término da próxima. O relatório deve enumerar as tarefas que foram planejadas mas não foram completadas, além de citar os motivos que levaram à não realização dessas tarefas.

O aluno que não entregar o relatório até a data e a hora definidas (**23h59 da data de término de uma iteração**) para o término de uma iteração será punido com a perda de **1,0 ponto** na nota final do trabalho.

7.Artefatos a Serem Entregues

O código fonte do jogo só precisará ser entregue na última iteração. Isto é, o código fonte deverá ser entregue ao professor até as **23h59** do dia **11/12/2016**, data da entrega do trabalho. A entrega deverá ser feita por meio do Ambiente de Aprendizagem On-line da PUC-Rio.

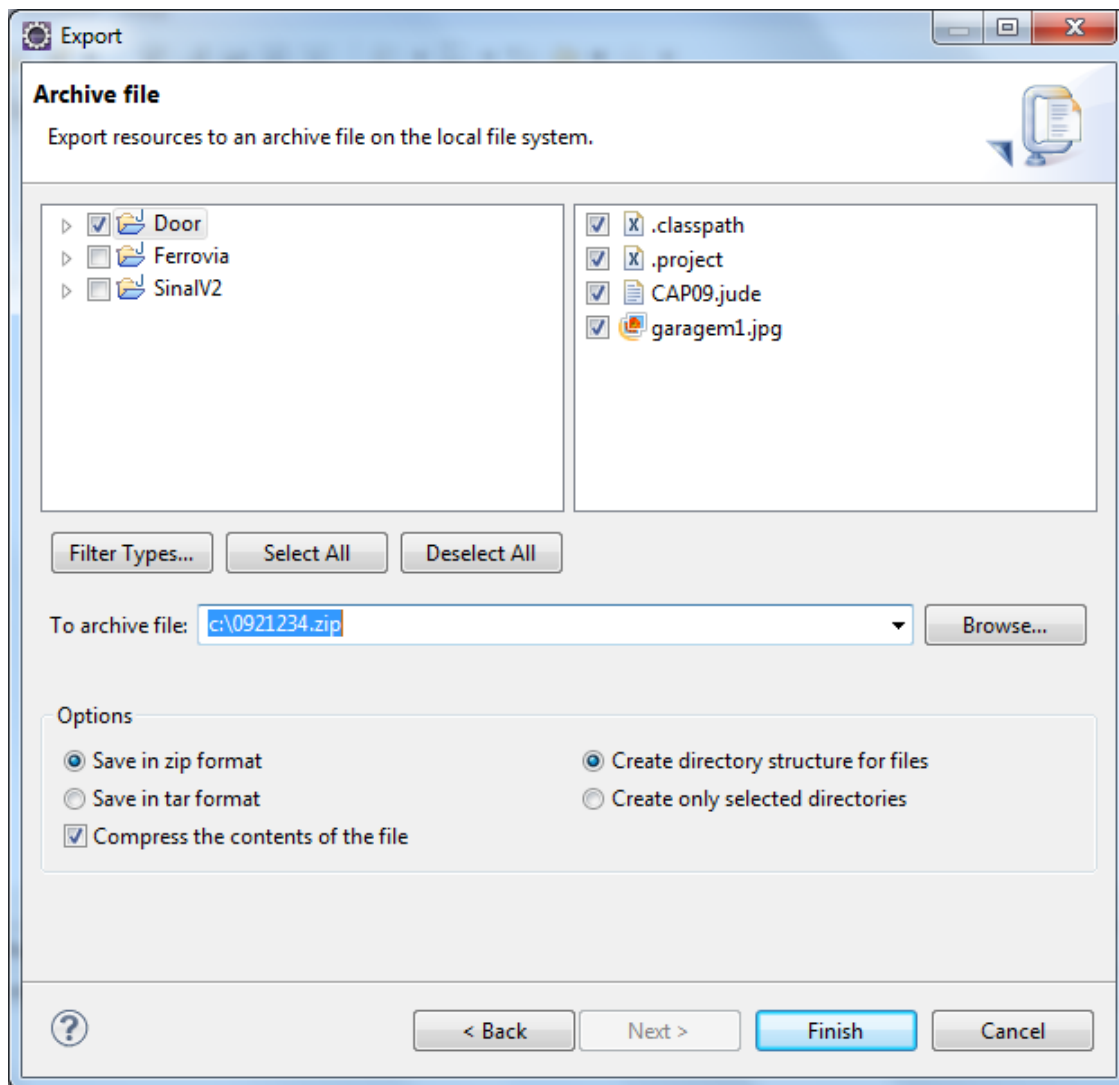
Juntamente com o 1º relatório, deve ser entregue um arquivo astah, ou jude, com um ou mais diagramas de classes e de pacotes resultantes da aplicação de engenharia reversa sobre o trabalho base. Esses diagramas irão ajudá-los a entender a arquitetura do trabalho base, tarefa fundamental para que se possa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

A segunda versão do modelo de classes do jogo deve ser entregue junto com o 2º relatório. Ele deve mostrar um esboço das classes que vocês irão inserir para viabilizar a comunicação dos clientes com o servidor. Crie, também, um ou dois diagramas de sequência que ilustrem alguns detalhes das interações responsáveis por essa comunicação.

Ao término da última iteração, 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 **TEM** de ser composto pela matrícula dos dois membros do grupo. Por exemplo, **1222345-1113456**.

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



8. Iterações – Proposta de Organização das Tarefas

1ª Iteração

Data de término: 19/11/2016

Tarefas: construir um chat que permita reuniões virtuais com um número qualquer de participantes. Iniciar a construção do servidor do jogo. Criar 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. Criar alguns diagramas de sequência que mostrem as interações necessárias ao funcionamento do servidor.

2ª Iteração

Data de término: 27/11/2016

Tarefas: concluir o servidor do jogo. Criar arquivo astah, ou jude, contendo as classes do trabalho base acrescidas das classes que irão viabilizar a comunicação dos clientes com o servidor. Criar alguns diagramas de sequência que mostrem as interações necessárias ao estabelecimento da comunicação do cliente com o servidor. Iniciar adaptação do trabalho base para funcionar como cliente do jogo.

3ª Iteração

Data de término: 04/12/2016

Tarefas: prosseguir com a adaptação do trabalho base para atuar como cliente do jogo. Iniciar a criação do diálogo inicial do jogo. Esse diálogo deve pedir ao jogador que forneça o nickname pelo qual ele será conhecido pelos outros jogadores e estabelecer a conexão inicial com o servidor. Após quatro jogadores terem se conectado, o servidor enviará mensagens para que os clientes iniciem o jogo. Caso o início não se dê em até três minutos contados a partir da primeira conexão, o programa cliente deverá enviar uma mensagem para o jogador e encerrar suas atividades.

4ª Iteração (última)

Data de término: 11/12/2016 (data de entrega)

Tarefas: concluir todos os componentes restantes.

9. Datas de Entrega e de Apresentações

1. A versão resultante da última iteração deverá ser entregue por meio do Ambiente de Aprendizagem On-line da PUC-Rio Isso terá de ser feito até as **23h59min** do dia **11/12/2016**. **Não haverá tolerância na entrega da versão final. Quem não entregá-la dentro do prazo definido receberá nota ZERO no G2.**
2. As apresentações ocorrerão nos dias **12, 13, 14, 15 e 16/12/2016**. **O aluno que faltar à sua apresentação receberá nota ZERO no G2.** A ordem e os horários das apresentações serão acertados com a turma na semana anterior às apresentações.

O trabalho deverá ser apresentado nas máquinas do LabGrad.

10. 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.