

### **Instruções Gerais Para o Trabalho:**

1. Esta atividade vale 20 pontos e poderá ser resolvida em grupo com no máximo 2 integrantes.
2. Caso você ache que falta algum detalhe nas especificações, você deverá fazer as suposições que julgar necessárias e escrevê-las no seu relatório. Pode acontecer também que a descrição dessa atividade contenha dados e/ou especificações supérfluas para sua solução. Utilize sua capacidade de julgamento para separar o supérfluo do necessário.
3. Para desenvolver esta atividade utilize preferencialmente as linguagens Python ou Java. Se quiser utilizar outra linguagem deverá justificar a sua escolha no seu relatório.
4. Como produtos da atividade serão gerados dois artefatos: códigos fontes da implementação e documentação da atividade.
5. Cada arquivo-fonte deve ter um cabeçalho constando as seguintes informações: nome(s) do(s) aluno(s), matrícula(s) e data.
6. O arquivo contendo a documentação da atividade (relatório) deve ser devidamente identificado com o(s) nome(s) e matrícula do(s) autor(es) do trabalho. O arquivo contendo o relatório deve, obrigatoriamente, estar no formato PDF.
7. Devem ser entregues os arquivos contendo os códigos-fontes e o arquivo contendo a documentação da atividade (relatório). Compacte todos os artefatos gerados num único arquivo no formato ZIP.
8. O prazo final para entrega desta atividade é até 23:59:00 do dia 22/06/2017.
9. O envio é de total responsabilidade do aluno. Não serão aceitos trabalhos enviados fora do prazo estabelecido.
10. Trabalhos plagiados serão desconsiderados, sendo atribuída nota 0 (zero) a todos os envolvidos.

## **1 Introdução**

O Segundo Trabalho Prático (TP) aborda o desenvolvimento (projeto e implementação) de uma versão do jogo Pacman particular idealizada pelo grupo. O jogo a ser desenvolvido tem alguns requisitos obrigatórios que devem ser cumpridos e outros adicionais que devem ser acrescentados pelo grupo durante a fase do projeto.

O Pacman é um *cult* na família dos *arcades* e você encontrará facilmente informações sobre o jogo na internet. Como sugestão, veja <https://en.wikipedia.org/wiki/Pac-Man>

### **1.1 Requisitos obrigatórios**

Os objetivos obrigatórios são:

1. O jogo deve implementar pelo menos cinco “elementos móveis” que atuarão simultaneamente na tela: um *pacman* (o come-come) e quatro *ghosts* (os fantasmas vermelho, azul, lilás e laranja). Cada um desses elementos será controlado por um autômato finito determinístico (AFD) executado como uma *thread* – um AFD e uma *thread* por elemento móvel.

2. Além dos elementos móveis, o jogo deve implementar um elemento imóvel chamado *árbitro*. O árbitro será responsável pelo gerenciamento e acompanhamento do jogo, atuando por exemplo para: criar e disparar as *threads*, colocar os *biscoitos* de tempos em tempos para o *pacman* comer (os *dots*, se o grupo escolher implementar essa opção), enviar os símbolos de entrada para cada AFD, etc. O árbitro também faz a interface do jogo com o usuário.
3. O *pacman* será comandado pelo usuário. O alfabeto para o pacman será as quatro teclas de movimentação: cima, baixo, esquerda e direita.
4. Os *ghosts* serão comandados automaticamente pelas *inteligências* dos AFDs – um AFD por *ghost*. O alfabeto para cada *ghost* faz parte das opções de projeto escolhidas pelo grupo, mas pode variar entre:
  - sinais temporais (tiks), enviados pelo árbitro de tempos em tempos para cada AFD.
  - sinais de posicionamento relativo, enviados pelo árbitro para cada AFD quando muda a posição do *pacman* em relação ao *ghost*: está na direita, na esquerda, em cima ou em baixo.
  - sinais de posicionamento absoluto, enviados pelo árbitro para cada AFD quando esse passa em alguma posição absoluta específica, por exemplo, alguma encruzilhada no labirinto.

As escolhas de cada ghost serão baseadas no desenho do AFD e nos sinais de entrada recebidos do árbitro que gerencia o jogo.

## 1.2 Requisitos opcionais

Propositamente, os requisitos para este trabalho não foram colocados de forma rígida. Cada grupo pode escolher implementar outros requisitos para tornar o jogo mais interessante e valorizar seu trabalho, ou mesmo modificar algum requisito “obrigatório” para tornar o jogo mais interessante – desde que a mudança não altere substancialmente a proposta do trabalho (utilização de AFDs) e seja aprovada antecipadamente pelo professor da disciplina.

Exemplos de requisitos opcionais podem ser: número variável de *ghosts*, oferta de “biscoitos” e/ou outros bônus para atrair o pacman, controle variável de dificuldade, etc. Seja criativo. Primeiro, garanta o necessário; depois, o céu dentro do prazo de entrega é o limite :)

## 2 Critérios de Correção

Serão adotados os seguintes critérios de correção para o trabalho:

1. **correção:** somente serão corrigidos códigos sem de erros de compilação;
2. **qualidade:** qualidade do projeto, comprometimento do grupo, apresentação e diversão do jogo;
3. **validação e verificação:** atendimento aos requisitos e execução correta conforme esperado;
4. **modularização e qualidade do código fonte:** legibilidade, indentação, etc.

Haverá uma apresentação oral e individual do trabalho.

Na ausência de plágio, as notas dos trabalhos corretos serão computadas individualmente da seguinte forma:  $\text{nota} = \text{nota\_apresentacao} * \text{nota\_trabalho}$ , ou seja, a nota final é ponderada pela nota da apresentação.

## 3 Alternativa para quem não está cursando POO

Conforme explicado em sala de aula, este trabalho foi idealizado para ser realizado em conjunto com a disciplina de POO. Atendendo aos alunos que não estão cursando POO, esses podem opcionalmente substituir o desenvolvimento do Pacman pelo trabalho de desenvolver os outros seguintes algoritmos (complementando o pacote do primeiro trabalho prático):

1. conversão ER  $\rightarrow$  AFN- $\lambda$
2. conversão AFN- $\lambda \rightarrow$  AFN
3. conversão AFN  $\rightarrow$  AFD
4. conversão AFD  $\rightarrow$  ER