

Modelagem e Simulação para jogos

Enunciado Trabalho do Grau B

Elaborar um jogo-protótipo a partir do desenvolvimento de um motor (*engine*) de simulação discreta estocástica.

O trabalho está dividido nas seguintes etapas:

1. Construção das classes e/ou funções que modelam processos de um modelo. Um processo pode ser modelado, por exemplo, através de uma função que implemente uma thread, sendo que esta pode receber objetos que serão retidos (thread é colocada para *sleep* por um certo tempo) pela função, e logo após decorrido este tempo alguma outra ação é realizada com o o objeto (ele pode ser destruído, por exemplo); Entre os elementos que devem ser modelados estão as filas, que terão objetos sendo inseridos e removidos a medida que a simulação é executada
2. Para testar o motor (estas classes desenvolvidas na etapa 1), deve ser construído um pequeno modelo-teste. Este modelo-teste corresponde a um posto de gasolina (com 1 processo que gera carros, um processo de atendimento (abastecimento), uma fila (de abastecimento) e uma saída. Este modelo-teste deve ser implementado com o motor desenvolvido e também modelado em MicroSaint, a fim de se comparar a execução deste mesmo modelo através destas duas abordagens diferentes. A comparação pode levar em conta o tamanho médio da fila de abastecimento, por exemplo. Se estes tamanho diferirem muito, isto pode sinalizar algum problema na implementação do motor.
3. No motor deve ser empregado o gerador pseudo-randômico LCG desenvolvido em sala de aula; Para fins de torná-lo thread-safe, é interessante empregar diferentes instâncias do gerador nos diversos pontos do modelo que precisam de sorteios.
4. Também devem ser implementadas as funções/métodos uniforme, exponencial e normal, que devem ser empregadas dentro do modelo. Estas funções devem ser testadas (via Qui_Quadrado) para verificação da qualidade do resultado que estão gerando. Estes testes podem ser feitos através da geração de sequencias de 100 valores que são em seguida contabilizados em classes para permitir em seguida a aplicação do teste do Qui-Quadrado.
5. Por fim, uma vez que o motor tenha sido validado (através da comapração com o modelo-teste) e que os geradores também tenham sido validados (via testes Qui_Quadrado), deve ser construído o protótipo do jogo baseado em um modelo que:
 - a. Possa ser pausado pelo jogador;
 - b. Assim que o estado do modelo sofrer alterações (isto é, quando o valor de alguma variável/atributo for alterado), as informações sobre o modelo deve ser reapresentadas para o jogador, ou seja, a tela do jogo deve ser *refreshed*;
 - c. Não é necessária uma apresentação gráfica/visual do modelo/jogo; a saída pode ser toda em console;
 - d. O papel do jogador é reparametrizar o modelo (isto é, alterar o valor de parâmetros do modelo, como quantidades de caixas atendendo, quantos atendentes estão operacionais, etc); toda vez que o usuário quiser reparametrizar, ele deve poder pausar

o jogo para poder efetuar estas alterações; uma outra possibilidade é permitir que o jogador faça estas alterações sem necessariamente parar o jogo, o que implica em incluir no laço do jogo a verificação de interação com o jogador.

Bom trabalho!!!