ODD Simulação de fluxo de fila

1. Propósito

Questão: Qual é o propósito deste modelo?

Resposta: O propósito do modelo é simular o fluxo de estudantes através de vários estágios em um sistema definido. A simulação permitirá analisar o comportamento da fila de atendimento de forma controlada e replicável, sem interferir diretamente no funcionamento real do restaurante.

Através da simulação, será possível testar diferentes cenários e estratégias de melhoria operacional, avaliando seus impactos antes de serem implementados na prática. A simulação possibilita a compreensão mais profunda dos fatores que influenciam os tempos de espera, como padrões de chegada dos comensais, capacidade de atendimento, *layout* do restaurante etc.

O objetivo principal do estudo das filas nesse restaurante é avaliar o potencial de aplicação da metodologia *Lean Thinking* para melhorar a eficiência operacional do restaurante, reduzindo os tempos de espera e proporcionando uma experiência mais satisfatória aos usuários. A simulação permitirá testar e comparar diferentes estratégias de melhoria. Ao final, espera-se que a simulação forneça subsídios concretos para a implementação de soluções efetivas, alinhadas com as necessidades e expectativas dos comensais. Basicamente o lean thinking é: identificação do gargalo/problema, identificação de sua causa raiz e resolução do problema com base na eliminação de desperdícios. Esses desperdícios são os mais diversos, como o tempo de espera ou a demora para realizar o pagamento, que geram os gargalos. Ou seja, os desperdícios geram os gargalos. Não precisa saber mais que isso sobre lean)

Além disso, o *NetLogo*, como ferramenta de modelagem e simulação, oferece recursos visuais para representar o comportamento da fila de forma clara e intuitiva. Isso facilita a compreensão dos resultados por parte de diferentes indivíduos, como gestores, funcionários e alunos, promovendo o engajamento e a tomada de decisões mais assertivas.

Portanto, a simulação da fila de atendimento utilizando o *NetLogo* apresenta-se como uma ferramenta importante para uma análise profunda e controlada dos fatores que impactam a experiência do usuário, contribuindo para uma gestão eficiente do restaurante e possibilitando a proposição de melhorias efetivas, reduzindo o risco de intervenções malsucedidas.

2. Entidades, variáveis de estado e escalas

Questão: Que tipos de entidades estão no modelo? Por quais variáveis ​​de estado, ou atributos, essas entidades são caracterizadas? Quais são as resoluções e extensões temporais e espaciais do modelo?

**Respostas:**

* **Entidades:**
  + **Estudantes:** Representam os indivíduos que passam pelos diferentes estágios do sistema.
  + **Patches:** Representam os diferentes estágios (ENTRADA, PAGAMENTO, B1 a B7, SAIDA) pelos quais os estudantes passam.
* **Variáveis de Estado:**
  + **Estudantes:**
    - location: O estágio atual onde o estudante se encontra.
    - time-in-stage: O tempo que o estudante passou no estágio atual.
    - total-time: O tempo total que o estudante passou desde o estágio ENTRADA.
  + **Patches:**
    - stage: Identificador do estágio representado pelo patch.
    - students-count: Contagem de estudantes presentes no estágio.
* **Temporal Resolution and Extent:**
  + O modelo avança em passos discretos de tempo (ticks).
  + Cada tick representa um pequeno incremento de tempo (ex: 1 minuto).
  + A simulação continua até que todos os estudantes tenham passado pelo estágio SAIDA.

**3. Process overview and scheduling**

**Questão:** Quem (ou seja, que entidade) faz o quê e em que ordem? Quando as variáveis de estado são atualizadas? Como o tempo é modelado, como passos discretos ou como um continuum onde ambos os processos contínuos e eventos discretos podem ocorrer?

**Resposta:**

* **Setup:** Inicializa a simulação, criando os patches e os estudantes.
* **Create-sector:** Define os patches que representam cada estágio.
* **Create-public:** Cria os estudantes e os posiciona no estágio inicial (ENTRY).
* **Go:** Função principal de execução que move os estudantes e exibe os tempos médios.
* **Move-students:** Lógica de movimento dos estudantes entre os estágios.
* **Update-stage-time:** Atualiza o tempo total e a contagem de estudantes em cada estágio.
* **Average-time:** Calcula o tempo médio para um dado estágio.
* **Display-average-times:** Exibe os tempos médios para todos os estágios.

**Pseudo-codigo para o agendamento:**

procedure setup

    create patches

    create students

    assign students to ENTRY stage

procedure go

    while not all students are in EXIT

        move-students

        update-stage-time

        display-average-times

        tick

procedure move-students

    for each student

        if student is not in EXIT

            move to next stage based on conditions

procedure update-stage-time

    for each stage

        update the total time and count of students

procedure display-average-times

    for each stage

        calculate and display average time spent

**4. Design concepts**

**Princípios básicos:** O modelo é baseado na teoria de filas e simulação de processos para observar fluxos em sistemas. Ele utiliza princípios de simulação discreta para modelar o movimento dos estudantes através dos estágios.

**Emergência:** O tempo médio gasto em cada estágio emerge das interações individuais dos estudantes com os estágios e do tempo que cada estudante leva para se mover de um estágio para outro.

**Adaptação:** Os estudantes se movem de um estágio para outro com base em regras predefinidas, que podem incluir condições específicas para a transição entre estágios.

**Objetivos:** O objetivo dos estudantes é passar por todos os estágios até alcançar o estágio EXIT.

**Aprendizagem:** O modelo não inclui aprendizado; os estudantes seguem regras fixas para mover entre estágios.

**Previsão:** Não aplicável, pois os estudantes não fazem previsões.

**Percepção:** Estudantes "percebem" o estágio atual em que estão e se movem para o próximo estágio com base nas regras de transição.

**Interação:** As interações são indiretas, através da ocupação de estágios pelos estudantes.

**Estocasticidade:** As transições entre estágios podem incluir elementos estocásticos, como a probabilidade de mover de um estágio para outro em cada tick.

**Coletivos:** Não há coletivos no modelo.

**Observação:** Os dados coletados incluem o tempo médio gasto em cada estágio e o tempo total de cada estudante. Esses dados são coletados e exibidos a cada tick.

**5. Inicialização**

**Questões:** Qual é o estado inicial do mundo do modelo, ou seja, no tempo t = 0 de uma execução de simulação? Em detalhes, quantas entidades de que tipo existem inicialmente, e quais são os valores exatos de suas variáveis de estado (ou como foram definidos estocasticamente)? A inicialização é sempre a mesma ou pode variar entre simulações? Os valores iniciais são escolhidos arbitrariamente ou com base em dados?  
**Resposta:**

* **Estado inicial:**
  + Um número fixo de estudantes é criado e colocado no estágio ENTRADA.
  + Todos os patches são definidos com seu respectivo estágio (ENTRADA, PAGAMENTO, B1 a B7, SAIDA).
  + A contagem de estudantes em cada estágio inicial é zero, exceto para o ENTRADA.

**6. Dados de entrada**

**Questão:** O modelo usa entrada de fontes externas, como arquivos de dados ou outros modelos, para representar processos que mudam ao longo do tempo?

**Resposta**: O modelo não utiliza dados de entrada externos para representar processos que mudam ao longo do tempo.

**7. Submodelos**

**Questões:** Quais são, em detalhes, os submodelos que representam os processos listados em 'Visão geral do processo e agendamento'? Quais são os parâmetros do modelo, suas dimensões e valores de referência? Como os submodelos foram projetados ou escolhidos, e como foram parametrizados e testados?  
**Resposta:**

* **Create-sector:** Define os patches que representam cada estágio. Os patches são coloridos para identificar visualmente cada estágio.
* **Create-public:** Cria um número fixo de estudantes e os posiciona no estágio ENTRY. Cada estudante é identificado por um número único.
* **Move-students:** Define a lógica de movimento dos estudantes entre os estágios. Cada estudante pode se mover para o próximo estágio com base em condições específicas, como uma probabilidade de transição.
* **Update-stage-time:** Atualiza o tempo total e a contagem de estudantes em cada estágio. Cada vez que um estudante se move, os tempos são atualizados para refletir a mudança.
* **Average-time:** Calcula o tempo médio que os estudantes gastam em cada estágio. A média é calculada dividindo o tempo total pelo número de estudantes que passaram pelo estágio.
* **Display-average-times:** Exibe os tempos médios para todos os estágios. Os resultados são mostrados na interface do NetLogo para fácil visualização.