

Projeto Integrador III

Sistema Autônomo

Caio Rosa, Felipe Moreti, Gabriel Pereira Pinto

Bacharelado em Ciência da Computação

Centro Universitário SENAC - Campus Santo Amaro (SENAC-SP)

Av. Engenheiro Eusébio Stevaux, 823 – Santo Amaro, São Paulo – CEP 04696-000 – SP – Brasil

caiolrosa1912@gmail.com, felipe.moreti7@gmail.com, gabrielpp@outlook.com



Resumo

Este poster busca descrever o processo de criação do Projeto integrado do terceiro semestre do Bacharelado em Ciência da Computação do Centro Universitário Senac. O projeto tem como objetivo criar e otimizar ao máximo um sistema autônomo que simule filas com uma grande quantidade de usuários. Onde os atendentes deste sistema façam trocas entre os seus postos de forma autônoma para melhor atender os usuários do mesmo.

1. Introdução e Objetivos

O objetivo desse projeto visa desenvolver um algoritmo para otimização de filas utilizando regras matemáticas ou regras de negócio. Faz se necessário que tenha interface gráfica para apresentar as filas e movimentação dos atendentes para melhor atender os diferentes postos em um menor tempo, contado em turnos, e apresentar um relatório ao final da execução mostrando todas as relações de tempo necessárias que ocorreram durante a execução do programa.

2. Metodologia

Tendo o problema em mãos foi iniciada então a pesquisa sobre ele. Em sua maior parte, o tema de "sistema autônomo" acaba se referindo a muitas outras áreas, em sua maioria sistemas de redes. No entanto durante a pesquisa foi encontrado a apostila do Prof. Dr. Norian Marranghello, onde ele trata sobre Redes Petri. Este conceito serviu de base para a criação dos conceitos do que viria a ser o trabalho. Durante sua explicação das noções fundamentais sobre a Rede Petri ele explica como ela pode ser dividida em 3 principais elementos, sendo estes os Estados, Ações e Relação de Fluxo.

"Estados são usados para modelar componentes passivos dos sistemas, isto é correspondem às suas variáveis de estado..."; "Ações são usadas para modelar os componentes ativos dos sistemas, ou seja, os eventos que levam o sistema de um estado a outro"; "Relação de fluxo, é usada para especificar como se dá a transformação de um estado em outro pela ocorrência de ações no sistema" (Marranghello, Noria, 2005). Desta forma foi possível perceber a correlação entre os componentes passivos como os Atendentes, as Ações como as Trocas a serem realizadas, Relação de fluxo seria o Fluxo de Usuários do sistema e por fim os Estados como a distribuição dos Atendentes nos guichês.

Assim, tendo em mente o conceito que era preciso criar foi escolhido utilizar de uma linguagem orientada a objeto para programar. Tendo a vista a necessidade de criar uma grande quantidade de funções e objetos para a execução do projeto isto viria a facilitar. Por questões gráficas e conhecimento prévio dos integrantes do grupo foi escolhido trabalhar com a ferramenta Unity, que por consequência fez com que c-sharp fosse a linguagem escolhida de programação e para um melhor desenvolvimento também foi utilizado o Visual Studio em conjunto. Então antes mesmo de começar a programar então foi escolhido estudar as bases de como seriam o código e a interface base do projeto

3. Desenvolvimento da base do projeto

Foi então iniciado a produção do projeto. O objetivo principal era fazer uma versão mais simples do projeto o quanto antes para assim poder continuar e melhorando conforme o andamento do projeto. Com isso em mente a primeira parte do código a ser implementada foi então a leitura dos dois arquivos que nos seriam disponibilizados para poder rodar o programa os arquivos de Setup e de Fila. O arquivo de Setup contém as informações principais para o funcionamento do sistema, sendo essas: O número total de atendentes; A quantidade total de postos de atendimento; A forma que estes atendentes estariam disponíveis e espalhados ao início do sistema; O valor do tempo da ação de

Troca; por fim as letras que indicam cada um dos postos com seus tempos individuais e os nomes do que aquele posto representa.

O arquivo de fila por outro lado contém os usuários, cada um deles tendo um determinado número que serve para identificarmos quem é aquele usuário no sistema e outro número para o tempo em que ele entra no sistema, junto disso quais são os postos que este usuário irá passar no sistema, por exemplo U1C2ABDE, onde o seu nome é U1, sua chegada no sistema é no tempo 2 e os postos pelo qual ele irá passar são A, B, D e E nesta ordem. Com ambas as leituras dos arquivos funcionando foi então possível testar o funcionamento do projeto, mesmo que este tenha sido feito apenas pelo console do visual studio.

4. Desenvolvimento do Protótipo

Após as devidas melhorias nas leituras e dados os testes feitos através do texto do console do visual studio, foi então iniciado a produção do que viria a ser o primeiro protótipo do trabalho.

Para o começo foi desenvolvido uma parte gráfica simples do trabalho, utilizando-se apenas de quadrados, cubos e botões simples do unity para poder verificar o funcionamento do trabalho. Com a leitura e o andamento da fase inicial do protótipo funcionando foi então o momento das criações das variáveis para implementação do relatório requisitado ao final de todo o processo do arquivo fila.

O relatório deveria ser simples, porém conter uma quantidade certa de informações com a precisão correta, tais informações requeridas eram: Tempo médio de um usuário dentro do sistema; Tempo médio de espera por um tipo de fila; Tempo médio por combinação de necessidades de atendimento, então se eu tivesse 3 usuários que passassem pelos mesmos postos, como por exemplo ABC, eu teria que ter o tempo médio desses 3 usuários no sistema e mais os tempos médios de todas as outras combinações presentes no arquivo de fila; por fim o usuário que passou o maior tempo dentro do sistema.

Com o relatório e a base principal do protótipo funcionando foi o momento de desenvolver a parte principal do projeto, o sistema de trocas, tal sistema tem como objetivo a realização de trocas entre os atendentes do sistema para que assim o atendimento possa ser o mais rápido possível, por exemplo, em algum determinado momento eu poderia ter 2 postos As, mas apenas um atendente neles, caso a fila comece a ficar muito grande é necessário que mais um atendente fique no segundo posto A, dessa forma é requerida uma troca. O desenvolvimento da troca tomou conta de uma boa parte do projeto, sendo que sua otimização perdurou até o final do mesmo.

5. Animações

Durante a realização do projeto foi observado pelo grupo que era necessário alguma forma de mostrar visualmente o que ocorria dentro do sistema para um melhor entendimento das funções do sistema mesmo para quem não o conhecesse. Desta forma foi optado por desenvolver pequenas animações para o este melhor entendimento.

Tais animações foram desenvolvidas utilizando o método Easing de animações em unity, o método consiste numa base de cálculos subsequentes para calcular as posições dos objetos de acordo com a sua velocidade, assim as imagens se reposicionam de acordo com a velocidade causando aos olhos humanos a ideia de movimento.

Com o código das animações funcionando elas foram implementadas em dois momentos dentro do sistema, em primeiro momento nos usuários quando eles se movimentam da primeira posição da fila em direção a posição que indica que eles estão sendo atendidos. E em outro momento durante a troca dos atendentes, para que assim fosse possível não apenas indicar a onde os atendentes existiam ou não como também seu ponto de partida e chegada durante a troca.

Com as animações funcionando e com um melhor entendimento do que ocorria dentro do sistema pequenas melhorias foram feitas no layout: primeiramente o local dos

atendentes foi passado para a parte inferior na tela (como é possível perceber da diferença entre as Imagens 2 e 3); depois foi incluído um reset automático ao fechar o relatório, que é gerado ao fim da leitura do arquivo fila, para não ser necessário fechar e abrir o programa novamente; por fim um contador que é utilizado para quando a fila se torna grande demais e não é mais possível ver os usuários do sistema, este acabou sendo implementado apenas na versão final mesmo que já fosse existente em código pois foi escolhido ter toda a identidade visual e a tipografia escolhida antes da implementação.

6. Identidade Visual

Com uma grande parte do código do projeto já funcionando foi trabalhado então a identidade visual do projeto para que fosse possível ter o melhor entendimento de tudo que ocorria dentro do sistema assim como detalhar o ambiente simulado no mesmo.

Foram escolhidas então as tipografias que seriam utilizadas no projeto a tipografia Roboto Light para os usuários e para o relatório ao final foi escolhida por ser mais fina e ocupar um menor espaço mas que fosse simples e de fácil leitura para passar as informações, e, a tipografia Sugarpunch para o contador de turnos, para o nome dos atendentes e para o contador da fila dos atendentes, esta foi escolhida para dar um maior destaque para essas informações ao mesmo tempo que fosse fácil a sua leitura. Para os atendentes e os usuários foi escolhido uma paleta de cores fortes sem grandes variações de tonalidade para dar individualidade dentro do agrupamento, mas não necessariamente uma relação exclusiva de personalidade, por isso foram trabalhados em vista de topo sem uma caracterização facial.

Para o ambiente do sistema foi idealizado fazer os balcões dos postos, as cadeiras dos atendentes e usuários sendo atendidos para manter a ideia de onde existem atendentes nos postos em funcionamento e onde os usuários estão esperando na fila ou sendo atendidos.

Para a criação, as cores escolhidas buscam contrastar com o plano de fundo, ou seja, o chão. Ainda assim, trazendo uma uniformidade entre os itens e os atendentes, gerando uma identidade semelhante para os dois.

Tendo em mãos esta parte da identidade visual ela foi então implementada no projeto utilizando-se de uma função de randomização para que os atendentes e usuários sofressem uma diversificação dentro do sistema.

7. Finalização do Projeto

Durante o processo final do projeto foram então implementadas e feitas diversas melhoras. Primeiramente foram criados a identidade visual da Interface de Usuário (UI) do projeto, esta foi criada baseando-se fora do que o sistema simula, pensando assim exatamente no usuário que utiliza o projeto. Foram estes os botões com as diferentes velocidades 1x, 2x, 3x, o botão de play e o botão de pause e a janela de relatório ao final. Em uma extensa fase de testes do sistema foi percebido um bug no sistema de trocas, assim foi necessário reestruturar e realizar novas otimizações do mesmo. Consequentemente foram corrigidos diversos bugs, detalhes menores da UI e da identidade visual. Com tudo feito foi então gerado o build final para a finalização do projeto.

Referências

- [1] MARRANGHELO, Norian. Redes de Petri: Conceitos e Aplicações. 33 p. Apostila – DCCE/IBILCE/UNESP, [S.l.], 2005. .
- [2] UNITY: Manual de Usuário. 5.6.
- [3] TUTORIAIS UNITY Interface de Usuário..
- [4] C-sharp *Guia de Programação*. .
- [5] VISUAL STUDIO *IDE- Guia de Usuário*. .