Desenvolvimento do jogo - POO II

Lucas Bonato Soares

Outubro 2024

Sumário

1 Mudança na forma como o jogo ocorre

Originalmente, questões embaralhadas aleatoriamente eram apresentadas para o jogador. Se o jogador acertar a resposta ele ganha um ponto, e se errar ele passa para a próxima questão sem ganhar pontos.

Agora, as questões não são mais embaralhadas, e não existe mais um sistema de pontos. O jogo agora é sobre o jogador tentar chegar na última questão, sendo que errar uma resposta faz com que tenha que recomeçar do zero.

1.1 Labels

Foram feitas adaptações em algumas labels para acomodar o novo sistema. Agora, a label de "Score 0/10"é um contador de questão atual. Ex: "Questão 1/10".

1.2 Variáveis

Variável score removida, pois o jogo não vai mais usar um sistema de pontuação.

A forma como o algoritmo trata a variável qNum foi alterada - Agora ela é usada como métrica para a label de questão atual.

1.3 Lógica

Havia um IF que verificava se o jogador acertou ou não para conferir o ponto de acordo, e depois do if o jogo ia pra próxima questão.

Agora, o IF verifica se o jogador errou ou não. Caso ele tenha errado, a função RestartGame() é chamada. Caso contrário, o jogo continua.

A função RestartGame() foi alterada - Ela não mais zera o score, e agora chama a janela GameOver.xaml.

O embaralhamento das questões foi removido na função StartGame(), que agora serve para chamar a janela Introducao.xaml.

1.4 Audio

Adicionado efeito sonoro ao errar uma questão (Tela GameOver.xaml) e música de vitória ao vencer o jogo (Venceu.xaml).

2 Novas janelas

Foi feita a criação de 3 novas janelas.

2.1 Introducao.xaml

Tela onde as instruções de como jogar residem. É a primeira tela que o jogador vê - antes mesmo da MainWindow. Fiz isso usando a função ShowDialog() na StartGame().

Ao fechar essa janela o jogo começa.

2.2 GameOver.xaml

Tela de fim de jogo quando o jogador erra uma resposta. Ela é chamada na função de RestartGame().

Ao fechar essa janela o jogo recomeça, mas sem mostrar a janela Introducao.xaml novamente.

2.3 Venceu.xaml

Tela de vitória quando o jogador acerta a última questão, parabenizando o jogador e anunciando o fim do jogo.

Nessa janela usei as classes SoundPlayer para tocar a música de vitória e DispatcherTimer para criar um efeito de "loop", onde a cada 1 segundo a cor do texto "PARABÉNS"muda de cor.

Ao fechar essa janela o jogo acaba e a aplicação fecha.

3 Questões

Cada questão foi alterada no quesito de conteúdo, mas algumas causaram alterações mais significativas no código da aplicação.

3.1 Questão 4

Pra fazer a cor do texto do botão mudar, tive que garantir que dentro do foreach no começo da função NextQuestion() exista uma linha que atribua a cor padrão do texto do botão, para assim poder alterar a cor especificamente na questão 4 sem atrapalhar nas outras. Usei as classes SolidColorBrush e ColorConverter para escolher as cores.

Além disso, eu quis deixar o texto do botão em negrito nessa questão em específico, então novamente adicionei no foreach uma linha que garante o FontWeight = normal, permitindo que eu altere o FontWeight para Bold apenas na questão 4 sem afetar as outras questões.

3.2 Questão 5 e 8

Criei alternativas que usam as variáveis do programa "número da questão atual"(qNum) e "total de questões"(questionNumbers.Count). Assim, essas alternativas são dinâmicas conforme o tamanho do quiz.

3.3 Questão 9

Criei uma alternativa que só possui duas respostas, escondendo os botões de baixo. Com isso, precisei novamente adicionar no foreach do começo da função NextQuestion() uma linha que garante que a visibilidade dos botões seja visível, para assim poder esconder os que queria nessa questão sem afetar o resto da aplicação.

3.4 Questão 11

Precisei alterar o fundo dos botões. Novamente, garantir que no foreach tenha a cor padrão, e daí altero a cor do fundo na questão especificamente.

3.5 Questão 12

Muitas mudanças foram feitas pra essa questão - é uma questão discursiva, ou seja, tem um textbox ao invés de 4 alternativas. Ainda adicionei um botão separado para "submit"da resposta digitada. Com isso, tive que garantir no começo da função NextQuestion() que essa textbox e esse botão de resposta discursiva sempre estejam escondidos, apenas aparecendo no caso especifico dessa questão.

Ainda tive que criar uma nova função onClick para o botão de envio da resposta discursiva: A função checkWrittenAnswer, e com essa função duas novas variáveis globais: WrittenAnswer e WrittenSubmission.

Essas duas variáveis string sempre estão com valor vazio, mas nessa questão, WrittenAnswer assume o valor da resposta correta enquanto WrittenSubmission assume o valor inserido pelo usuário. Com isso, a função checkWrittenAnswer verifica a equidade das duas variáveis para decidir se o jogador acertou ou não.

3.6 Questão 13

Para essa questão, eu fiz uma pegadinha onde nenhuma das 4 alternativas responde corretamente o enunciado. A resposta está no número da questão, exibido no canto superior esquerdo, que corresponde a resposta certa.

Para isso, criei um botão escondido (chamado de ans5) no local da resposta. Esse botão apenas tem Visibility.Visible durante esta questão, e ele é o único botão com Tag = 1 (significando que é a resposta certa) durante essa questão.

Esse botão não tem fundo ou borda, sendo invisível até que o mouse passe por cima dele.

3.7 Questão 14

Para a questão 14, eu novamente fiz uma pegadinha. Dessa vez, a pergunta "você está preparado?"só tem como resposta Sim ou Não, mas quando o jogador passa com o mouse por cima da alternativa ela troca para a inversa. Ou seja, bem quando o jogador vai clicar em "Sim", ela troca para "Não".

Para fazer isso eu tive que criar duas variáveis novas para cada botão: ConteudoOriginal e ConteudoAlternativo. O conteudo original é o valor que o botão

tem inicialmente na tela, enquanto o conteúdo alternativo é o que ele assume quando o mouse passa por cima dele.

Com isso, fiz também duas novas funções mouseEnter e mouseLeave que são compartilhadas por todos os botões. Elas verificam o valor das duas novas variáveis e troca elas, mas apenas se a questão atual utiliza dessa caracteristica.

3.8 Em suma

- Novo tratamento para poder trocar a cor do texto dos botões.
- Novo tratamento para poder trocar o estilo do texto dos botões (Bold, Normal, etc).
- Novo tratamento para poder trocar a cor do fundo dos botões.
- Uso das variáveis de questão atual e total de questões nas alternativas.
- Novo tratamento para permitir a omissão de alternativas.
  - Por ex.: Deixar apenas 2 alternativas em uma questão.
- Suporte para criação de questões discursivas.
- Suporte para criação de novos botões.
  - Suporte para botões escondidos.
- Novo comportamento de "Hover"do mouse em botões.

4 Gabarito das questões

- Questão 1 - XLIX
- Questão 2 - ←
- Questão 3 - 2:51
- Questão 4 - ESSA (quarta alternativa)
- Questão 5 - 15
- Questão 6 - ©
- Questão 7 - Não há resposta
- Questão 8 - 8
- Questão 9 - Dromedário
- Questão 10 - Não existe
- Questão 11 - Essa (terceira alternativa)

- Questão 12 - 42
- Questão 13 - 13: Deve-se clicar no número da questão atual lá em cima
- Questão 14 - Botão da direita (muda para SIM quando hover do mouse)
- Questão 15 - 2