Частное учреждение образования

«Колледж бизнеса и права»

ОТЧЕТ ПО ТЕМЕ №1

Т.091005

Руководитель практики Е.Н.Коропа

Учащийся А.О.Зеневич

2022

**«WPF. ТЕСТ»**

**ВАРИАНТ 4**

Условие

Мортал комбат: https://meduza.io/quiz/fataliti-ili-brutaliti

Задача: разработать WPF-приложение по шаблону теста (указан по ссылке) из

индивидуального задания. Приложение должно содержать:

* Начальное окно с описанием и кнопкой «Начать»
* Текстовое поле вопроса
* Блоки вариантов ответа
* Кнопки перехода к следующему вопросу и/или вывода результата
* Дополнительные функции теста согласно выбранного варианта

(таймер/очки/изображения и др.)

Механика теста должна быть соблюдена!

Предусмотреть возникновение ошибок.

Изображения, сложность компоновки, бутоны, скроллбары и другие «дизайнерские»

решения также оцениваются.

Чем оригинальнее реализация теста, тем выше оценка.

Код программы

MainWindow.xaml

<Window x:Class="MortalKombatTest.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:MortalKombatTest"

mc:Ignorable="d"

Title="Mortal Kombat Quiz" Height="900" Width="1000" WindowStartupLocation="CenterScreen">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Margin" Value="5"/>

<Setter Property="Padding" Value="10"/>

<Setter Property="FontSize" Value="16"/>

<Setter Property="Width" Value="150"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Text="Mortal Kombat Quiz" FontSize="28" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Grid.Row="1" Text="Тест на знание игры Mortal Kombat. Прошло уже больше 25 лет с момента выхода первой части. " FontSize="16" HorizontalAlignment="Center"/>

<StackPanel Grid.Row="2" HorizontalAlignment="Center">

<TextBlock x:Name="QuestionText" FontSize="24" TextWrapping="Wrap" Margin="0 0 0 20"/>

<ItemsControl x:Name="AnswerList">

<ItemsControl.ItemTemplate>

<DataTemplate>

<Button Content="{Binding}" Click="AnswerButton\_Click"/>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</StackPanel>

<StackPanel Grid.Row="3" HorizontalAlignment="Center">

<Button x:Name="StartButton" Content="Начать тест" Click="StartButton\_Click" Height="96" Width="233"/>

<StackPanel Orientation="Horizontal" Height="234">

<Button x:Name="FinishButton" Content="Завершить тест" Click="FinishButton\_Click" IsEnabled="False" Height="80"/>

<Button x:Name="NextButton" Content="Следующий вопрос" Click="NextButton\_Click" IsEnabled="False" Height="80"/>

</StackPanel>

<TextBlock x:Name="ScoreText" Text="Очки: 0" FontSize="16" Margin="0 20 0 0"/>

<TextBlock x:Name="TimeText" Text="00:00" FontSize="16" Margin="0 10 0 0"/>

</StackPanel>

</Grid>

</Window>

MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Threading;

namespace MortalKombatTest

{

public partial class MainWindow : Window

{

private Test \_test;

private DispatcherTimer \_timer;

public MainWindow()

{

InitializeComponent();

}

private void StartButton\_Click(object sender, RoutedEventArgs e)

{

List<Question> questions = new List<Question>

{

new Question("1992 год. Выходит первая часть Mortal Kombat. " +

"По сюжету вам нужно сразиться в жестоком турнире " +

"под названием «Смертельная битва». Как зовут " +

"организатора и главного босса?",

new List<string>{"Шан Цзун", "Шао Кан", "Шан Као Цзун"}),

new Question("1993 год. Во второй части Mortal Kombat Шан Цзун терпит " +

"поражение от Лю Кана, затем просит своего повелителя " +

"Шао Кана сохранить ему жизнь и предлагает план захвата " +

"Земного царства. Здесь же мы встречаем новых персонажей. " +

"Например, Джакса. Вспомните, кто он?",

new List<string>{"Он из спецназа США", "Бывший космонавт",

"Чемпион по американскому футболу", "Тренер по фитнесу"}),

new Question("В 1995 году выходит легендарная Mortal Kombat 3. " +

"В ней много нововведений. Например, игрокам дают " +

"возможность бегать, выбрасывать противников на " +

"другую арену, добавляют финальное комбо. Какое?",

new List<string>{"Animality", "Babality", "Friendship", "Fatality"}),

new Question("Вслед за игрой в 1995 году снимают фильм «Смертельная битва» " +

"по мотивам игры Mortal Kombat. Кого из персонажей игры нет в фильме?",

new List<string>{"Шиннок", "Лю Кан", "Соня Блейд", "Китана"}),

new Question("В сентябре 1997 года на PlayStation и на ПК выходит Mortal Kombat 4. " +

"Что становится главным дополнением в этой части?",

new List<string>{"Возможность использовать оружие, разбросанное по арене",

"Экран «Choose your Destiny», позволяющий выбрать противников",

"Возможность использовать Fatality, а потом сразу Brutality",

"Восточная тема в игре"}),

new Question("И вот мы уже в 2011 году. Выходит Mortal Kombat 9, которая " +

"представляет собой одновременно продолжение и перезагрузку франшизы. " +

"Кто из новых персонажей появляется в этой части?",

new List<string>{"Фредди Крюгер", "Ночной Волк", "Джейсон Вурхиз", "Кожаное Лицо"}),

new Question("2015 год. Выходит десятая часть в серии файтингов — Mortal Kombat Х " +

"(произносится как «Мортал Комбат Икс»). Сильно меняется геймплей, " +

"появляются восемь новых персонажей, игра становится доступна на Android и iOS. " +

"И впервые в истории Mortal Kombat охватывает большой промежуток времени. " +

"Сколько лет длится эта битва по сюжету?",

new List<string>{"25 лет", "5 лет", "99 лет", "300 лет"}),

};

\_test = new Test(questions);

\_test.NextQuestion();

UpdateUi();

\_timer = new DispatcherTimer

{

Interval = TimeSpan.FromSeconds(1)

};

\_timer.Tick += Timer\_Tick;

\_timer.Start();

StartButton.IsEnabled = false;

StartButton.Visibility = Visibility.Collapsed;

NextButton.IsEnabled = true;

FinishButton.IsEnabled = true;

}

private void UpdateUi()

{

QuestionText.Text = \_test.CurrentQuestion.QuestionText;

AnswerList.ItemsSource = \_test.CurrentQuestion.Answers;

ScoreText.Text = $"Очки: {\_test.Score}";

}

private void AnswerButton\_Click(object sender, RoutedEventArgs e)

{

var answer = (sender as Button)?.Content.ToString();

\_test.CheckAnswer(answer);

if (\_test.IsFinished)

{

Finish();

}

else

{

\_test.NextQuestion();

UpdateUi();

}

}

private void NextButton\_Click(object sender, RoutedEventArgs e)

{

\_test.NextQuestion();

UpdateUi();

}

private void FinishButton\_Click(object sender, RoutedEventArgs e)

{

Finish();

}

private void Finish()

{

\_timer.Stop();

MessageBox.Show($@"Тест завершен! Количество очков: {\_test.Score} {Environment.NewLine}Время: {\_test.TimeElapsed:mm\:ss}");

Close();

}

private void Timer\_Tick(object sender, EventArgs e)

{

\_test.TimeElapsed += TimeSpan.FromSeconds(1);

TimeText.Text = $@"Время: {\_test.TimeElapsed:mm\:ss}";

}

}

}

public class Test

{

private readonly List<Question> \_questions;

private int \_currentQuestionIndex;

public Test(List<Question> questions)

{

this.\_questions = questions;

\_currentQuestionIndex = -1;

Score = 0;

TimeElapsed = TimeSpan.Zero;

IsFinished = false;

}

public Question CurrentQuestion => \_questions[\_currentQuestionIndex];

public int Score { get; private set; }

public TimeSpan TimeElapsed { get; set; }

public bool IsFinished { get; private set; }

public void NextQuestion()

{

if (\_currentQuestionIndex < \_questions.Count - 1)

{

\_currentQuestionIndex++;

}

else

{

IsFinished = true;

}

}

public void CheckAnswer(string answer)

{

if (CurrentQuestion.IsCorrect(answer))

{

Score++;

}

}

}

public class Question

{

public string QuestionText { get; }

public List<string> Answers { get; }

public string CorrectAnswer { get; }

public Question(string questionText, List<string> answers)

{

QuestionText = questionText;

Answers = answers;

CorrectAnswer = Answers[0]; // верный ответ

}

public bool IsCorrect(string answer)

{

return answer == CorrectAnswer;

}

}

Результат программы



