**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Отчет по домашнему заданию

Проект «МАФИЯ»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Корчевский Александр |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2020 г.

**Описание проекта:**

Разработать WPF приложение для автоматизации игры в мафию. Оно должно помогать ведущему с самого начала игры: настроить количество людей и оптимальное количество ролей для них, рандомно распределить роли, отсчитывать время на каждого человека и роль во все игровые фазы (день и ночь). Интерфейс игры должен быть интуитивно понятен и иметь стиль.

**Диаграмма классов:**



**Текст программы:**

**MainWindow.xaml**

<Window x:Class="MAFIA.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:fa5="http://schemas.fontawesome.com/icons/"

xmlns:vm="clr-namespace:MAFIA.ViewModels"

xmlns:sys="clr-namespace:System;assembly=mscorlib"

xmlns:local="clr-namespace:MAFIA"

mc:Ignorable="d"

Title="{Binding Path=Title}"

Icon="C://MAFIA/icon.jpg"

Background="WhiteSmoke"

Height="450" Width="700">

<Window.Resources>

<DataTemplate DataType="{x:Type vm:SettingsViewModel}">

<Grid Margin="10">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="100"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Border CornerRadius="9" Grid.ColumnSpan="2" Margin="0, 5, 10, 5" Background="AntiqueWhite" BorderBrush="DimGray" BorderThickness="2" />

<Border CornerRadius="9" Grid.ColumnSpan="2" Grid.Row="1" Margin="0, 5, 10, 5" Background="AntiqueWhite" BorderBrush="Black" BorderThickness="2" />

<Border CornerRadius="9" Grid.ColumnSpan="2" Grid.Row="2" Margin="0, 5, 10, 5" Background="AntiqueWhite" BorderBrush="Black" BorderThickness="2" />

<Border CornerRadius="9" Grid.ColumnSpan="2" Grid.Row="3" Margin="0, 5, 10, 5" Background="AntiqueWhite" BorderBrush="Red" BorderThickness="2" />

<Border CornerRadius="9" Grid.ColumnSpan="2" Grid.Row="4" Margin="0, 5, 10, 5" Background="AntiqueWhite" BorderBrush="Red" BorderThickness="2" />

<Border CornerRadius="9" Grid.ColumnSpan="2" Grid.Row="5" Margin="0, 5, 10, 5" Background="AntiqueWhite" BorderBrush="Red" BorderThickness="2" />

<Label VerticalAlignment="Center" Margin="10" Content="Игроки: "/>

<TextBox VerticalAlignment="Center" Name="PlayersNum" Margin="0, 10, 10, 10" Grid.Column="1" Height="22" Width="150" Text="{Binding PlayersNumber, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" />

<Label VerticalAlignment="Center" Grid.Row="1" Margin="10" Content="Мафия: "/>

<TextBox VerticalAlignment="Center" Grid.Row="1" Margin="0, 10, 10, 10" Grid.Column="1" Height="22" Width="150" Text="{Binding MafiasNumber, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"/>

<Label VerticalAlignment="Center" Grid.Row="2" Margin="10" Content="Дон мафии: "/>

<ComboBox SelectedIndex="{Binding DonNumber, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" Margin="0, 10, 10, 10" Grid.Column="1" VerticalAlignment="Center" Grid.Row="2" Height="22" Width="150">

<ComboBoxItem Content="0"/>

<ComboBoxItem Content="1"/>

</ComboBox>

<Label VerticalAlignment="Center" Grid.Row="3" Margin="10" Content="Шериф: "/>

<ComboBox SelectedIndex="{Binding SheriffNumber, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" Grid.Column="1" Margin="0, 10, 10, 10" VerticalAlignment="Center" Grid.Row="3" Height="22" Width="150">

<ComboBoxItem Content="0"/>

<ComboBoxItem Content="1"/>

</ComboBox>

<Label VerticalAlignment="Center" Grid.Row="4" Margin="10" Content="Доктор: "/>

<ComboBox SelectedIndex="{Binding DoctorNumber, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" Grid.Column="1" Margin="0, 10, 10, 10" VerticalAlignment="Center" Grid.Row="4" Height="22" Width="150">

<ComboBoxItem Content="0"/>

<ComboBoxItem Content="1"/>

</ComboBox>

<Label VerticalAlignment="Center" Grid.Row="5" Margin="10" Content="Мирные жители: "/>

<TextBox VerticalAlignment="Center" Grid.Row="5" IsReadOnly="True" Margin="0, 10, 10, 10" Grid.Column="1" Height="20" Width="150" Text="{Binding CitizensNumber, UpdateSourceTrigger=PropertyChanged}"/>

<Button Command="{Binding NextWindowTemplateCommand, UpdateSourceTrigger=PropertyChanged}" Grid.Column="2" Grid.RowSpan="6" Background="Beige">

<fa5:ImageAwesome HorizontalAlignment="Center" VerticalAlignment="Center" Icon="Solid\_Forward" Height="50" Foreground="BurlyWood"/>

</Button>

</Grid>

</DataTemplate >

<DataTemplate DataType="{x:Type vm:PickingViewModel}">

<Grid Margin="10">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="100"/>

</Grid.ColumnDefinitions>

<Label Content="{Binding LabelContext, UpdateSourceTrigger=PropertyChanged}" Foreground="{Binding FontColor, UpdateSourceTrigger=PropertyChanged}" FontSize="100" HorizontalAlignment="Center" VerticalAlignment="Center"></Label>

<Image Grid.Column="1" Source="{Binding CardPath, UpdateSourceTrigger=PropertyChanged}" />

<Button Command="{Binding NextPlayerCardCommand}" Grid.Column="2" FontSize="30" Background="Beige" ToolTip="Click here to see next role!">

<fa5:ImageAwesome HorizontalAlignment="Center" VerticalAlignment="Center" Icon="Solid\_Forward" Height="50" Foreground="BurlyWood"/>

</Button>

</Grid>

</DataTemplate>

<DataTemplate DataType="{x:Type vm:RandomizeViewModel}">

<DockPanel>

<Label DockPanel.Dock="Top" HorizontalAlignment="Center" FontSize="40" Margin="10" Content="Номера для игроков: "/>

<Button Margin="10" DockPanel.Dock="Bottom" Height="200" Command="{Binding NextWindowTemplateCommand, UpdateSourceTrigger=PropertyChanged}" Background="Beige">

<fa5:ImageAwesome HorizontalAlignment="Center" VerticalAlignment="Center" Icon="Solid\_Forward" Height="100" Foreground="BurlyWood"/>

</Button>

<Label HorizontalAlignment="Center" FontSize="50" Content="{Binding RandomString, UpdateSourceTrigger=PropertyChanged}"/>

</DockPanel>

</DataTemplate>

<DataTemplate DataType="{x:Type vm:RoundViewModel}">

<DockPanel Margin="10">

<StatusBar DockPanel.Dock="Bottom">

<StatusBarItem>

<fa5:ImageAwesome Icon="Solid\_Stopwatch" Margin="0, 2, 2, 2" Height="20"/>

</StatusBarItem>

<StatusBarItem>

<Label FontSize="14" FontWeight="Bold" Content="{Binding StopWatch}"/>

</StatusBarItem>

<Separator Margin="5, 0, 2, 0" Foreground="Black" BorderThickness="1" BorderBrush="Black"/>

<StatusBarItem>

<fa5:ImageAwesome Icon="Regular\_ThumbsUp" Height="20"/>

</StatusBarItem>

<StatusBarItem>

<Label HorizontalAlignment="Center" FontSize="14" FontWeight="Bold" Content="{Binding ElectedPlayers, UpdateSourceTrigger=PropertyChanged}"/>

</StatusBarItem>

<Separator Margin="5, 0, 5, 0" Foreground="Black" BorderBrush="Black" BorderThickness="1"/>

<StatusBarItem>

<fa5:ImageAwesome Icon="Solid\_UserFriends" Height="20"/>

</StatusBarItem>

<StatusBarItem >

<WrapPanel>

<Label Content="{Binding BlackNumber, UpdateSourceTrigger=PropertyChanged}" FontSize="14" FontWeight="Bold"/>

<Label Content="VS" FontSize="14" FontWeight="Bold"/>

<Label Content="{Binding RedNumber, UpdateSourceTrigger=PropertyChanged}" Foreground="Red" FontSize="14" FontWeight="Bold"/>

</WrapPanel>

</StatusBarItem>

<StatusBarItem HorizontalAlignment="Right">

<Label Content="{Binding Round, UpdateSourceTrigger=PropertyChanged}" FontSize="14" FontWeight="Bold"/>

</StatusBarItem>

</StatusBar>

<Grid>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">

<Label HorizontalAlignment="Center" FontSize="{Binding HeaderSize, UpdateSourceTrigger=PropertyChanged}" Content="{Binding Header, UpdateSourceTrigger=PropertyChanged}"/>

<Label HorizontalAlignment="Center" FontSize="{Binding HeaderSize, UpdateSourceTrigger=PropertyChanged}" Content="{Binding UnderHeader, UpdateSourceTrigger=PropertyChanged}"/>

<WrapPanel Margin="0,0, 0, 20" HorizontalAlignment="Center">

<Button IsEnabled="{Binding PlayButtonIsEnabled, UpdateSourceTrigger=PropertyChanged}" Command="{Binding StopWatchStartCommand, UpdateSourceTrigger=PropertyChanged}" Height="50" Width="50" Margin="0, 0, 15, 0">

<fa5:ImageAwesome Icon="Solid\_Play" Height="25"/>

</Button>

<Button IsEnabled="{Binding PauseButtonIsEnabled, UpdateSourceTrigger=PropertyChanged}" Command="{Binding StopWatchPauseCommand, UpdateSourceTrigger=PropertyChanged}" Height="50" Width="50" Margin="15, 0, 15, 0">

<fa5:ImageAwesome Icon="Solid\_Pause" Height="25"/>

</Button>

<Button Command="{Binding NextPlayerCommand, UpdateSourceTrigger=PropertyChanged}" Height="50" Width="50" Margin="15, 0, 0, 0">

<fa5:ImageAwesome Icon="Solid\_Forward" Height="25"/>

</Button>

</WrapPanel>

<WrapPanel HorizontalAlignment="Center">

<Label Content="{Binding StatusBar, UpdateSourceTrigger=PropertyChanged}" FontSize="20" FontWeight="Bold"/>

<TextBox IsEnabled="{Binding CouldBeSelected, UpdateSourceTrigger=PropertyChanged}" Text="{Binding SelectedPlayerNumber, UpdateSourceTrigger=PropertyChanged, Mode=TwoWay}" Height="26" FontSize="20" FontWeight="Bold" Width="100"/>

</WrapPanel>

</StackPanel>

</Grid>

</DockPanel>

</DataTemplate>

<DataTemplate DataType="{x:Type vm:TimerViewModel}">

<DockPanel>

<Label FontSize="40" Content="{Binding TimeLabel, UpdateSourceTrigger=PropertyChanged}"/>

</DockPanel>

</DataTemplate>

</Window.Resources>

<Window.DataContext>

<vm:MainWindowViewModel/>

</Window.DataContext>

<ContentControl Content="{Binding CurrentViewModel}"/>

</Window>

**PickingViewModel.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using MAFIA.ViewModels.Base;

using MAFIA.Infrastructure.Commands;

using System.Reflection.Emit;

using System.Windows.Input;

using System.Windows;

using System.Windows.Data;

using MAFIA.Models;

namespace MAFIA.ViewModels

{

class PickingViewModel : ViewModel

{

private int \_PlayerNumber;

public int PlayerNumber

{

get => \_PlayerNumber;

set => Set(ref \_PlayerNumber, value);

}

private string \_LabelContext;

public string LabelContext

{

get => \_LabelContext;

set => Set(ref \_LabelContext, value);

}

private string \_CardPath;

public string CardPath

{

get => \_CardPath;

set => Set(ref \_CardPath, value);

}

private string \_FontColor;

public string FontColor

{

get => \_FontColor;

set => Set(ref \_FontColor, value);

}

#region Команды

public ICommand NextPlayerCardCommand { get; }

private bool CanNextPlayerCardCommandExecuted(object p) => true;

private void OnNextPlayerCardCommandExecuted(object p)

{

if (++PlayerNumber == GameDeck.deck.Length)

action.Invoke(GameDeck);

else

{

LabelContext = "#" + (PlayerNumber + 1);

CardPath = GameDeck.deck[PlayerNumber].Path;

FontColor = GameDeck.deck[PlayerNumber].Color;

}

}

#endregion

private readonly Action<Deck> action;

private Deck \_GameDeck;

public Deck GameDeck

{

get => \_GameDeck;

set => Set(ref \_GameDeck, value);

}

public PickingViewModel(Deck gameDeck, Action<Deck> ParentAction)

{

action = ParentAction;

GameDeck = gameDeck;

NextPlayerCardCommand = new LambdaCommand(OnNextPlayerCardCommandExecuted, CanNextPlayerCardCommandExecuted);

PlayerNumber = 0;

LabelContext = "#" + (PlayerNumber + 1);

CardPath = GameDeck.deck[PlayerNumber].Path;

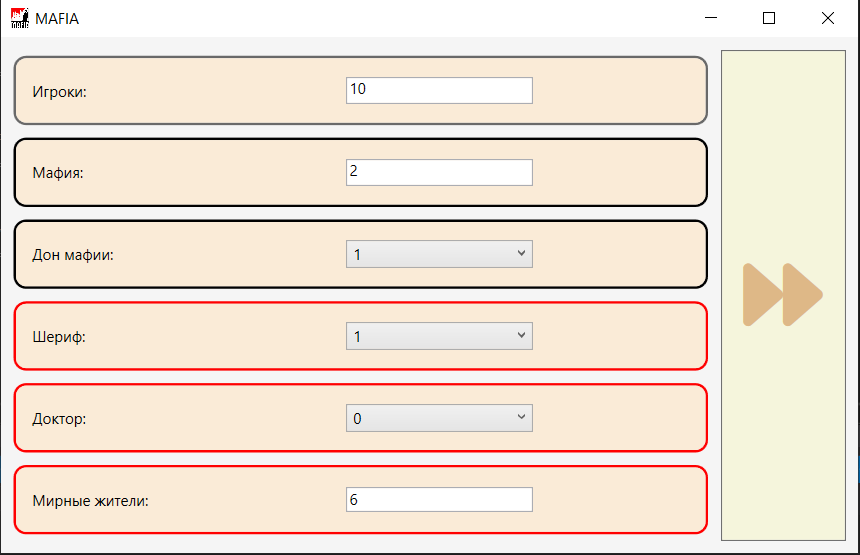
FontColor = GameDeck.deck[PlayerNumber].Color;

}

}

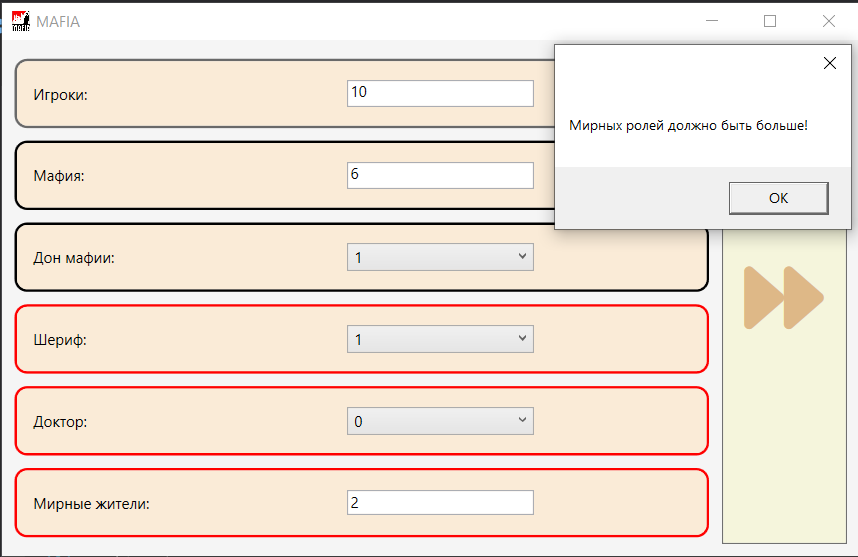
}

**Результаты:**

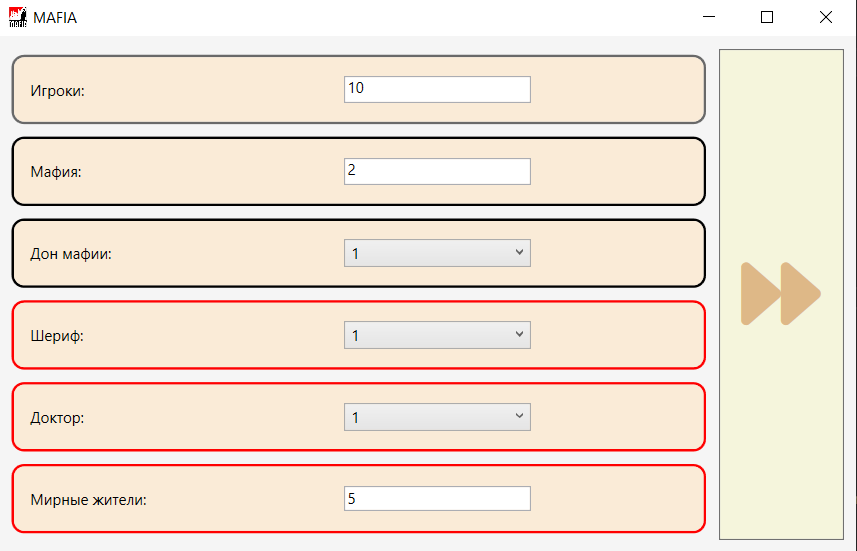


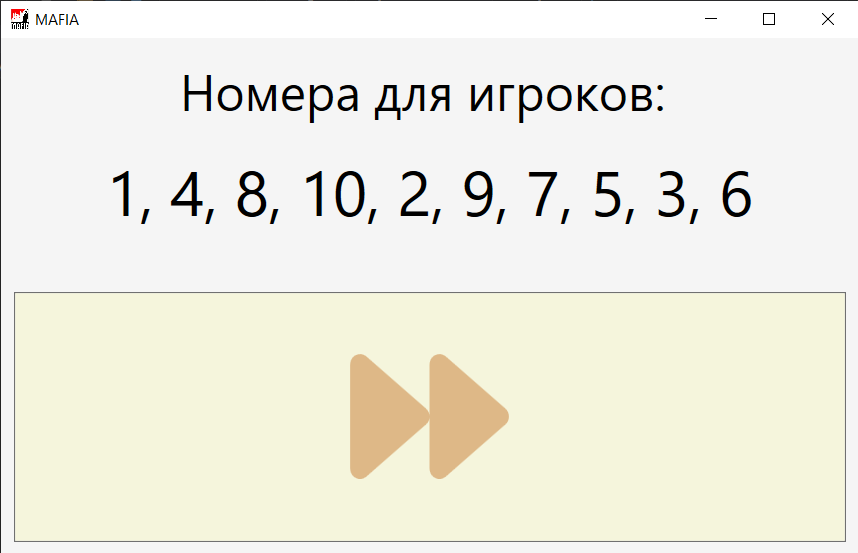
Приложение встречает пользователя меню с настройкой игрового лобби. На игровом сленге роли, находящиеся на стороне мирного города, называют красными, остальные роли, носящие негативный характер воздействия на мирных, - черными соответственно. Данный нюанс находит отражение в цвете рамочки вокруг роли.

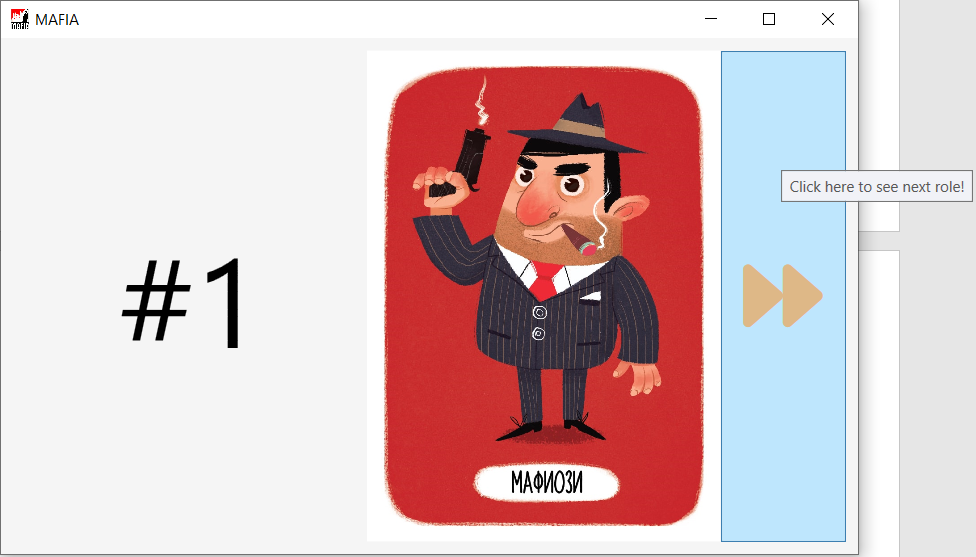
У пользователя есть возможность менять любые поля, кроме мирных жителей, оно высчитывается автоматически. Также игра следит за общим соотношением красных и черных ролей и не даст запустить игру с неправильными настройками.

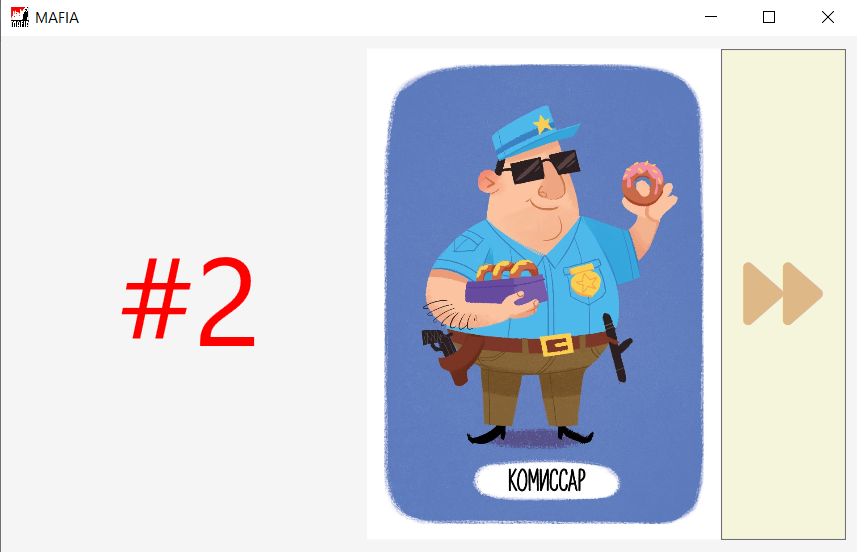


После правильной настройки ролей происходит раздача номеров игрокам. С помощью этой последовательности ведущий может поменять рассадку игроков, она играет важную роль в игровом процессе.

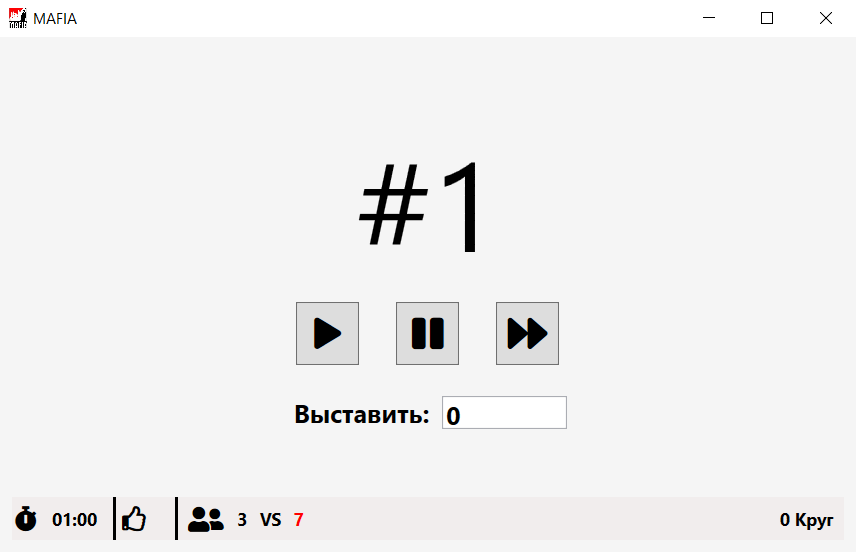
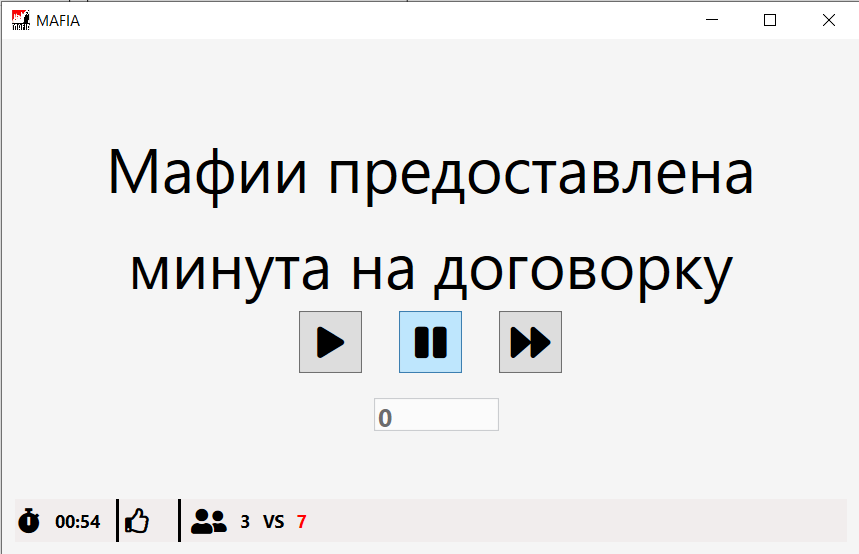




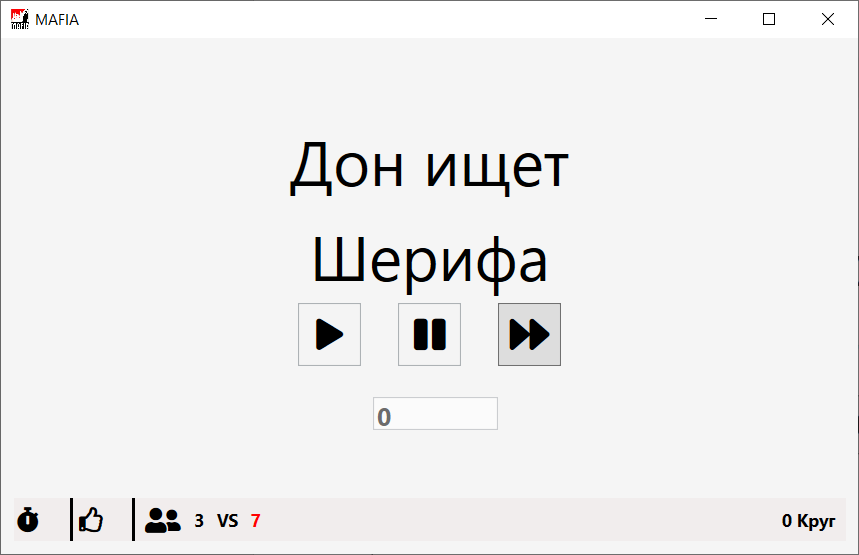
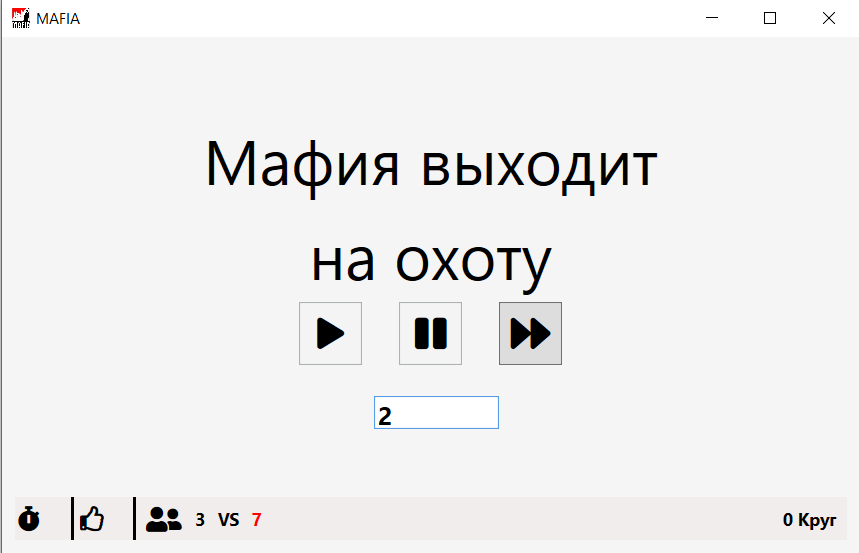
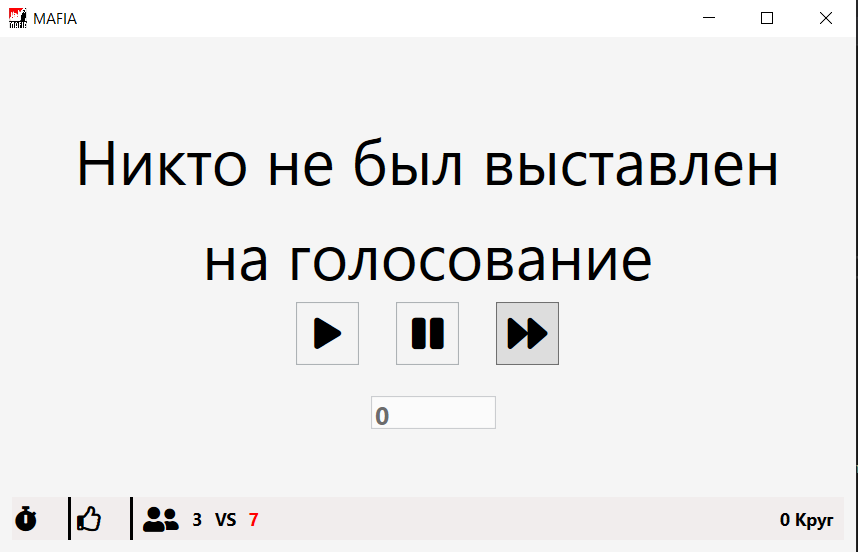
Затем производится рандомная раздача карточек: 

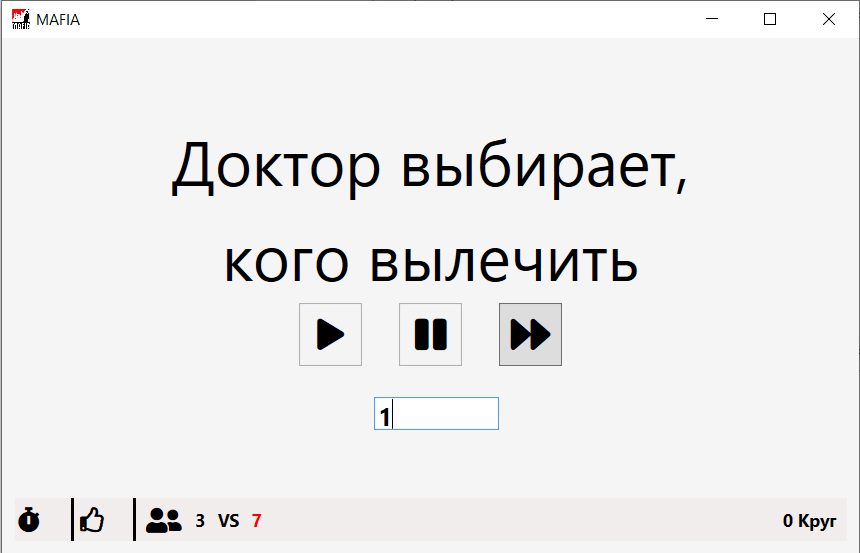


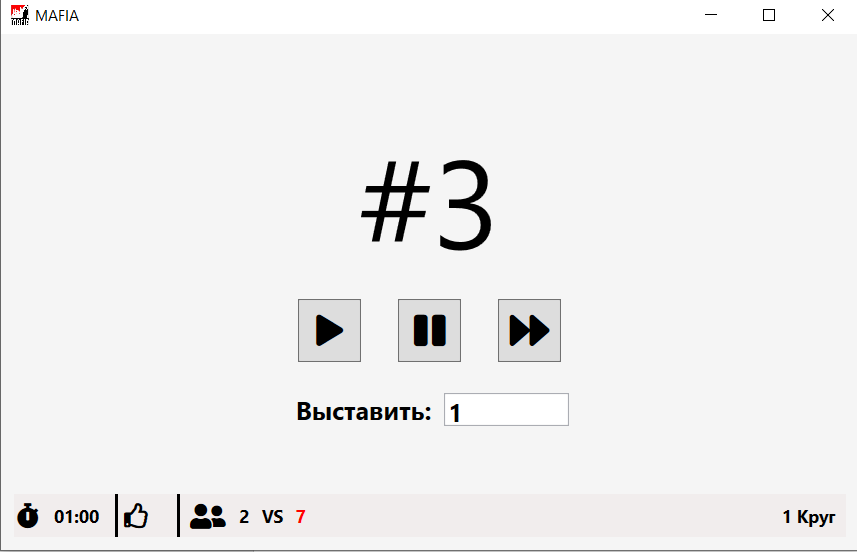
После раздачи ролей нас встречает основное меню приложения. Нижний тулбар содержит игровую информацию: оставшееся время, выставленные на голосование игроки, соотношение черных и красных игроков и номер круга. Доступные кнопки управления: запуск/остановка времени и переход на следующий этап игры.



Поле того, как пройдет время каждого игрока, и никто не был выставлен, наступает ночь.

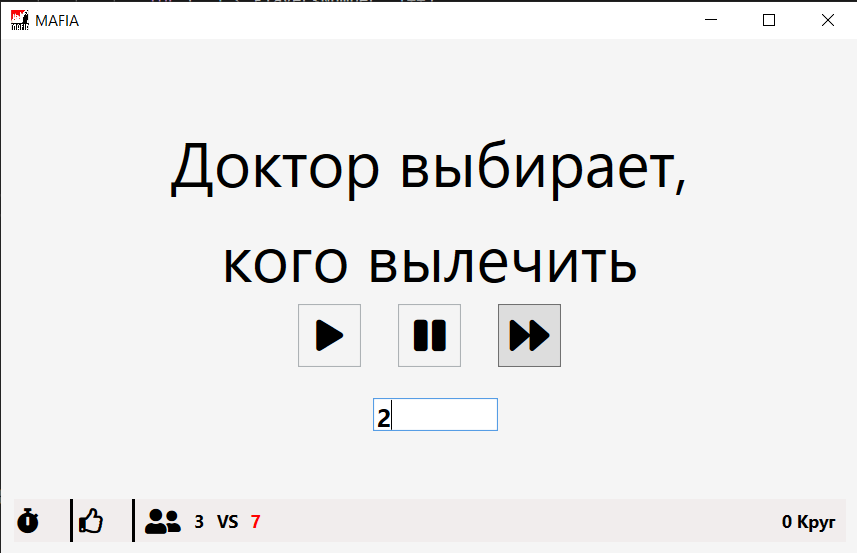


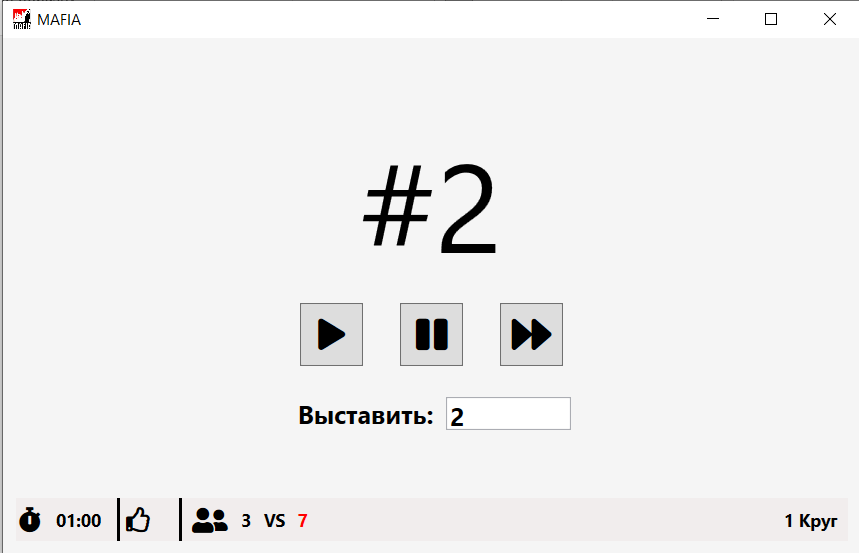




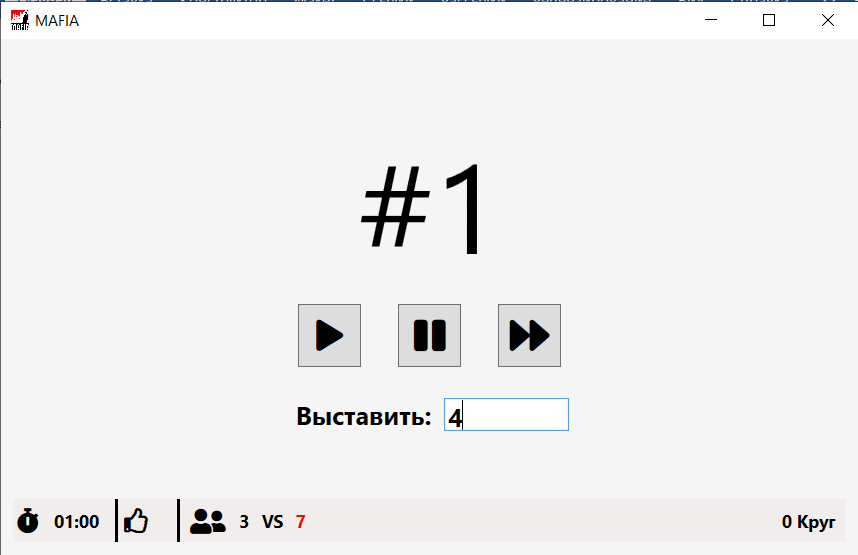
Новый круг должен начинаться со следующего игрока, после открывающего предыдущего круга. Второй игрок отстрелен, а доктор не попал, следовательно начинается круг с игрока под номером 3. По тулбару мы можем видеть, что мафия попала в одного из своих, т.е. черных игроков стало меньше.

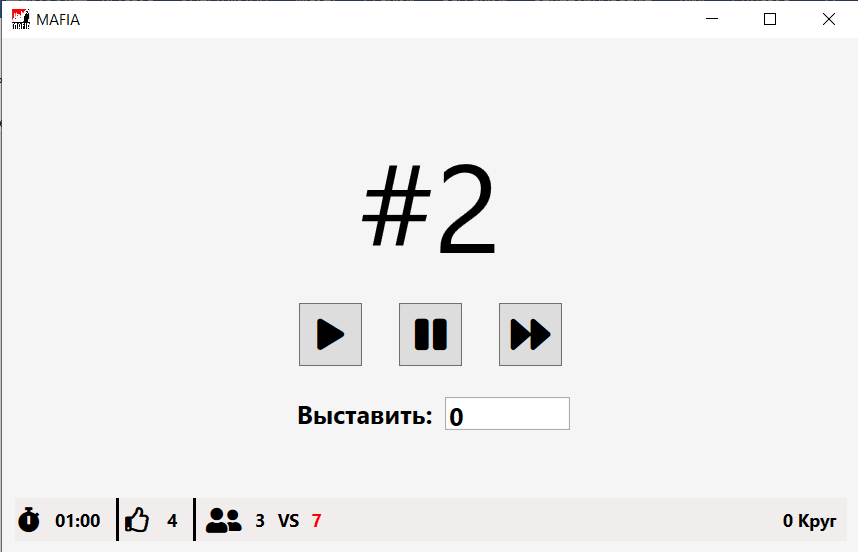
Если доктор угадывает, 1 круг начнется со второго игрока, а общее число игроков не изменится:





Также есть возможность выставлять игроков на голосование.





После выставления игрока, он записывается в тулбар для ведущего.

**Вывод**

Во время разработки проекта я обрел большой опыт в использовании WPF и практике языка программирования C#. Также приложение построено по шаблону MVVM и следует его концепции на каждом этапе. Также одним из решений было использование векторных изображений, но не в качестве <Image/>, а в реализации их в стиле, схожем с шрифтами. Для этого потребовалось установить в проект пакет FontAwesome5, его использование сильно облегчило использование иконок.

