Topic of the project: “The Maze” game

Annotation: We build a maze game, where the user has to move from start to finish through the walls without touching them as fast as possible. The database is created for keeping information about games history and results, which can be shared to social network using API.

Members of the team and their roles:

Osmanova Yana 146 - GUI, application logic
Badaeva Elina 146 - DB, application logic

Central repository address: https://github.com/d1stort/TheMaze.git

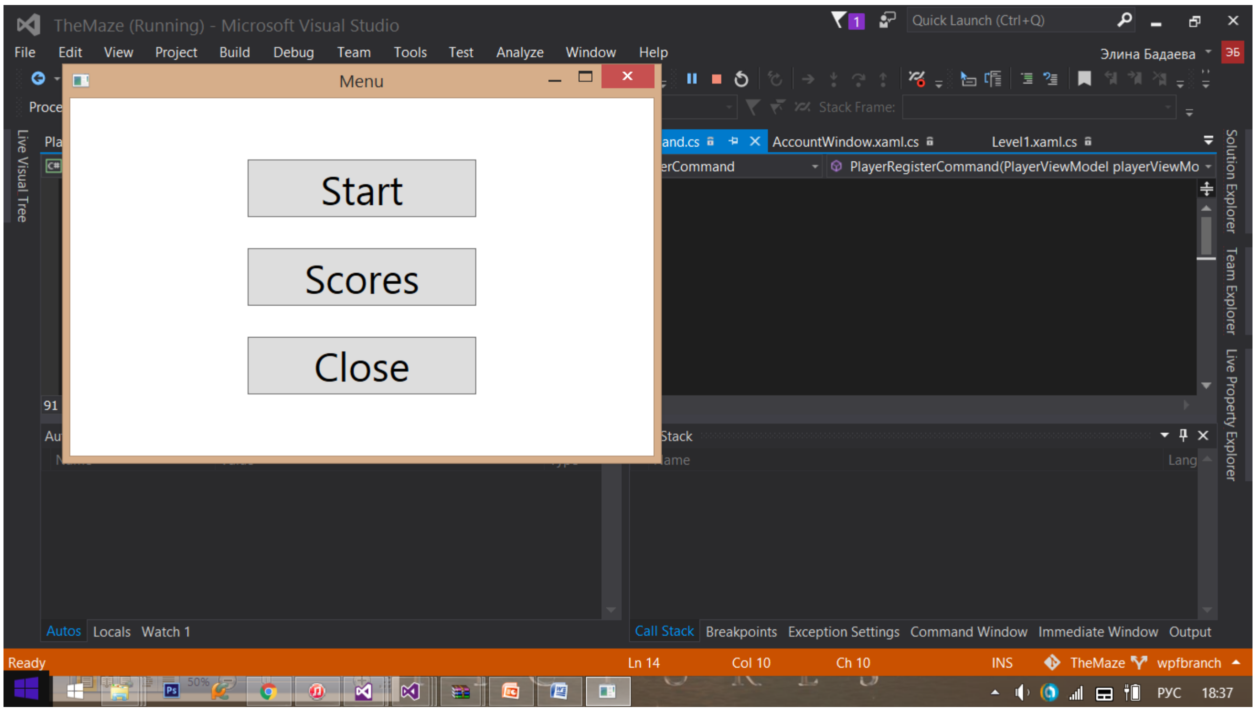
List of classes:

1. `public partial class MainWindow - (private void closeButton_Clicked, private void startButton_Clicked, private void statsButton_Clicked, static IQueryable<Player> LoadPlayersFromDB, private void ShowPlayers, private void goBackButton_Clicked) allows to control the main logic of application. For instance, it has methods for handling clicked button events.`
2. `class PlayerViewModel - is the bridge between the view and the model. The ViewModel retrieves data from the Model and manipulates it into the format required by the View. It notifies the View if the underlying data in the model is changed, and it updates the data in the Model in response to UI events from the View.`
3. `public partial class RegistrationWindow : Window - (private void CloseRegWindButton_Click, public RegistrationWindow()) allows to register your nickname in case you have not already have an account.`
4. `public partial class LoginWindow : Window - (private void PlayButtonLW_Click, public LoginWindow()) allows to log in to the application for keeping player’s game history.`
5. `public partial class Level1 : Window - (private static extern bool SetCursorPo, private static void SetCursor, public Level1(), private void mouseOver_finish, private void mouseEnter_label, private void Level_Loaded, private void BackTMenu_Click) allows to move the mouse pointer from the start to the finish, touching any of the walls is banned.`
6. `public partial class AccountWindow : Window - (public AccountWindow(),private void YesButtonAc_Click, private void NoButtonAc_Click) allows to understand whether you have an account or not. If you have, the LoginWindow will be opened. Otherwise, the RegistrationWindow will be opened.`
7. `public class Player - (private void OnPropertyChanged) is initial domain class with entities. It provides a view-independent representation of entities. The design of the model is optimized for the logical relationships and operations between entities.`
8. `class Context: DbContext - (public Context() : base("PlayersDb"), public DbSet<Player> Players, protected override void OnModelCreating)`
9. `internal sealed class Configuration : DbMigrationsConfiguration<TheMaze.Context> - (public Configuration(), protected override void Seed) derives from DBContext class and exposes DbSet properties.`
10. `internal class PlayerUpdateCommand : ICommand - (public PlayerUpdateCommand, public bool CanExecute, public event EventHandler CanExecuteChanged, public void Execute) includes methods for turning the LoginButton active and for processing activities in windows.`

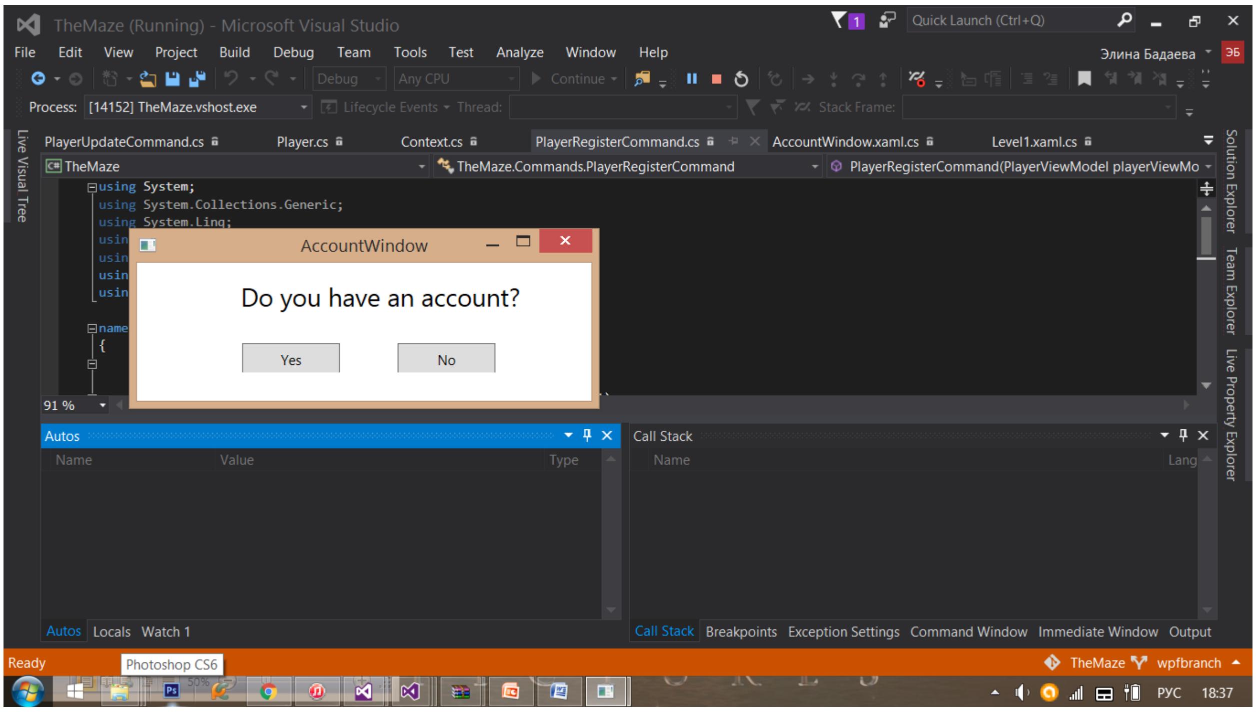
11. `internal class PlayerRegisterCommand : ICommand - (public PlayerRegisterCommand, public bool CanExecute, public event EventHandler CanExecuteChanged, public void Execute) includes methods for turning the RegisterButton active and for processing activities in windows.`

Program interface:

MainWindow allows selecting one of three in the menu: Start, Scores, Close

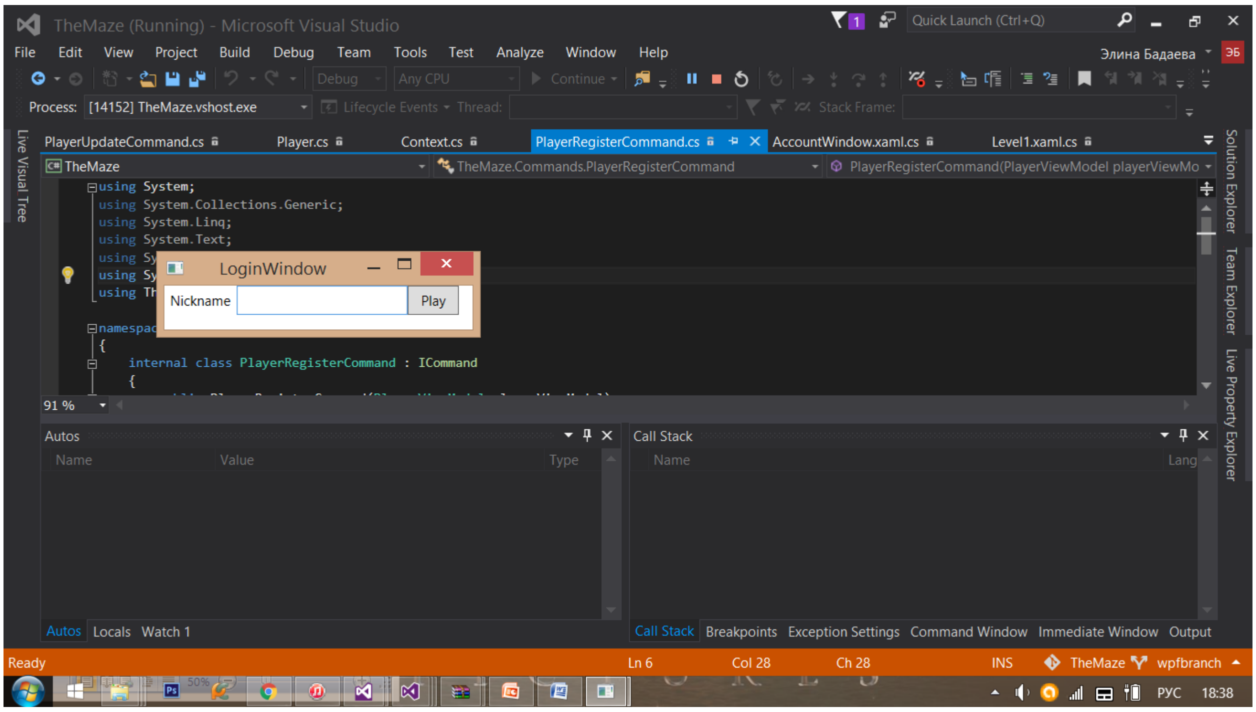

TheMaze (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Menu
Start
Scores
Close
AccountWindow.xaml.cs
Level1.xaml.cs
PlayerRegisterCommand(PlayerViewModel playerViewMo
Autos Locals Watch 1
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output
Ready
Ln 14 Col 10 Ch 10 INS
TheMaze wpfbranch

After selecting Start-button you are asked whether you have an account or not.


TheMaze (Running) - Microsoft Visual Studio
Process: [14152] TheMaze.vshost.exe
PlayerUpdateCommand.cs Player.cs Context.cs PlayerRegisterCommand.cs AccountWindow.xaml.cs Level1.xaml.cs
TheMaze
TheMaze.Commands.PlayerRegisterCommand
AccountWindow
Do you have an account?
Yes
No
Autos
Call Stack
Ready

```
using System;
using System.Collections.Generic;
using System.Linq;
```

If yes:

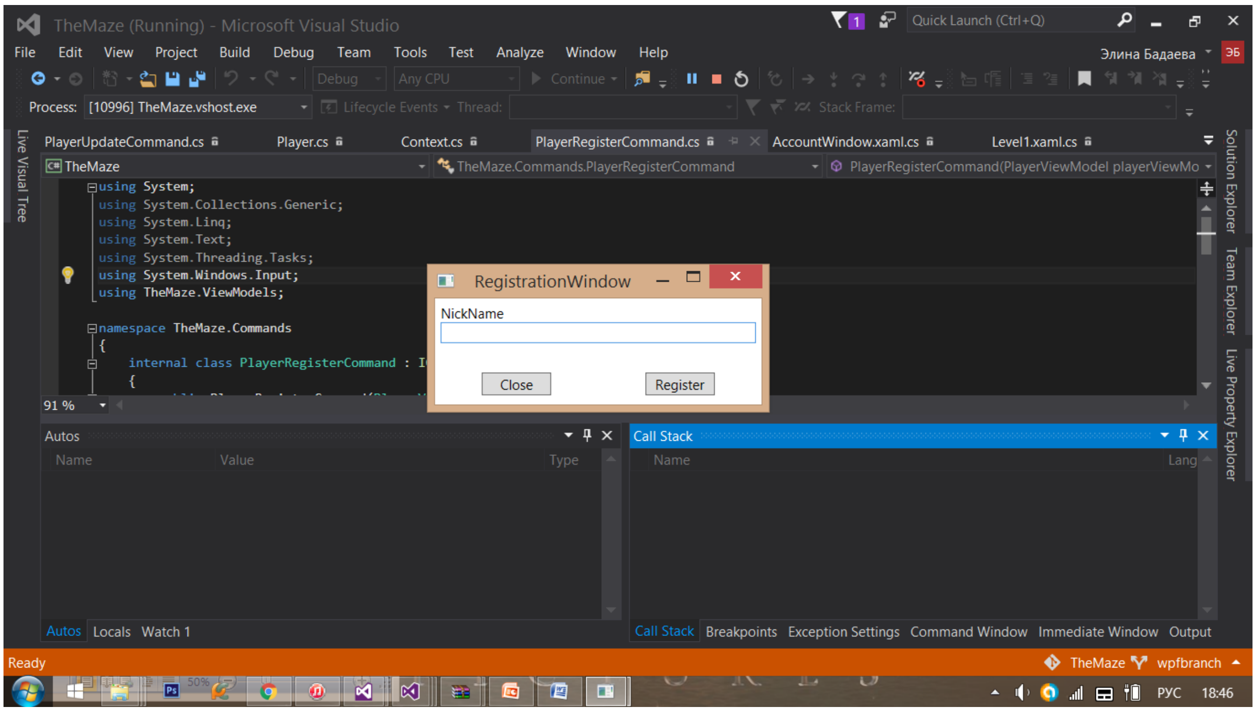


TheMaze (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Process: [14152] TheMaze.vshost.exe
PlayerUpdateCommand.cs Player.cs Context.cs PlayerRegisterCommand.cs AccountWindow.xaml.cs Level1.xaml.cs
TheMaze.Commands.PlayerRegisterCommand
PlayerRegisterCommand(PlayerViewModel playerViewMo
LoginWindow
Nickname
Play
Autos
Call Stack
Ready
Ln 6 Col 28 Ch 28 INS
18:38

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Sy
using Sy
using Th

namespac
{
    internal class PlayerRegisterCommand : ICommand
    {
```

If no:

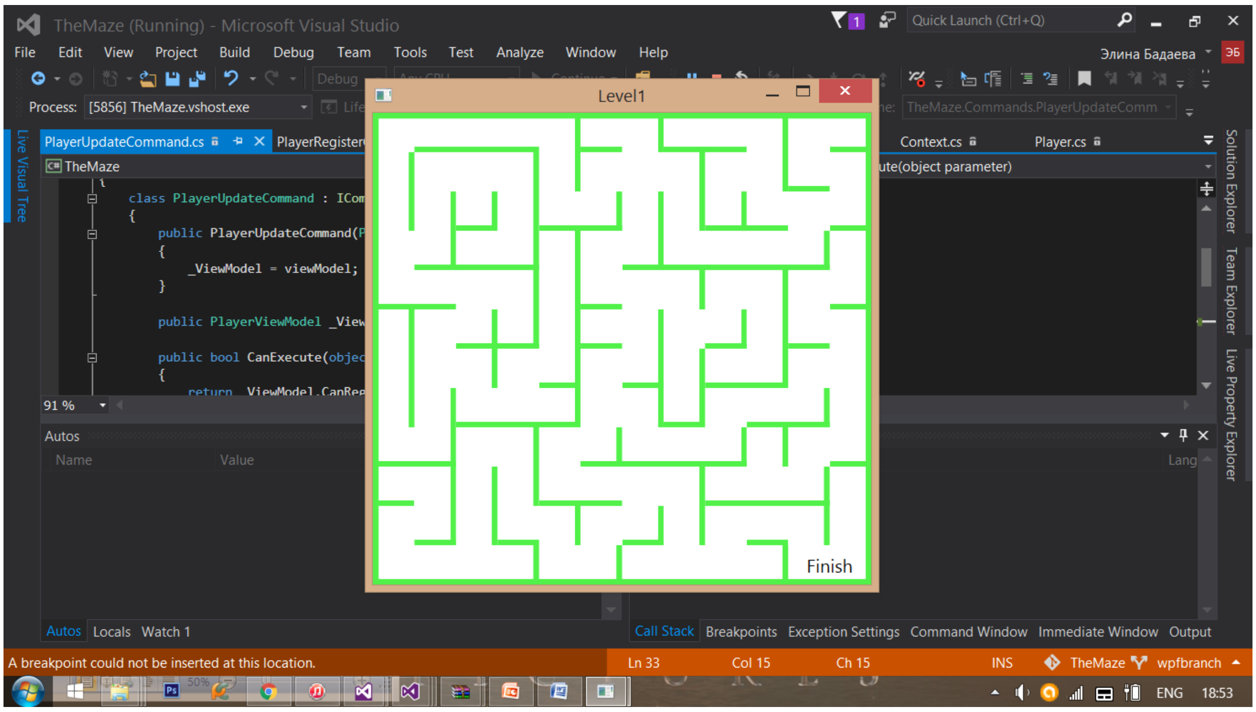


TheMaze (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Process: [10996] TheMaze.vshost.exe
PlayerUpdateCommand.cs Player.cs Context.cs PlayerRegisterCommand.cs AccountWindow.xaml.cs Level1.xaml.cs
TheMaze.Commands.PlayerRegisterCommand
PlayerRegisterCommand(PlayerViewModel playerViewMo
RegistrationWindow
NickName
Close
Register
Autos
Call Stack
Ready
18:46

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Input;
using TheMaze.ViewModels;

namespace TheMaze.Commands
{
    internal class PlayerRegisterCommand : I
    {
```

After successful logging in or registration you are suggested to try the first Level of the game.


TheMaze (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Process: [5856] TheMaze.vshost.exe
PlayerUpdateCommand.cs
PlayerRegister
Context.cs
Player.cs
Level1
Finish
Autos
Name Value
Autos Locals Watch 1
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output
Ln 33 Col 15 Ch 15 INS
ENG 18:53

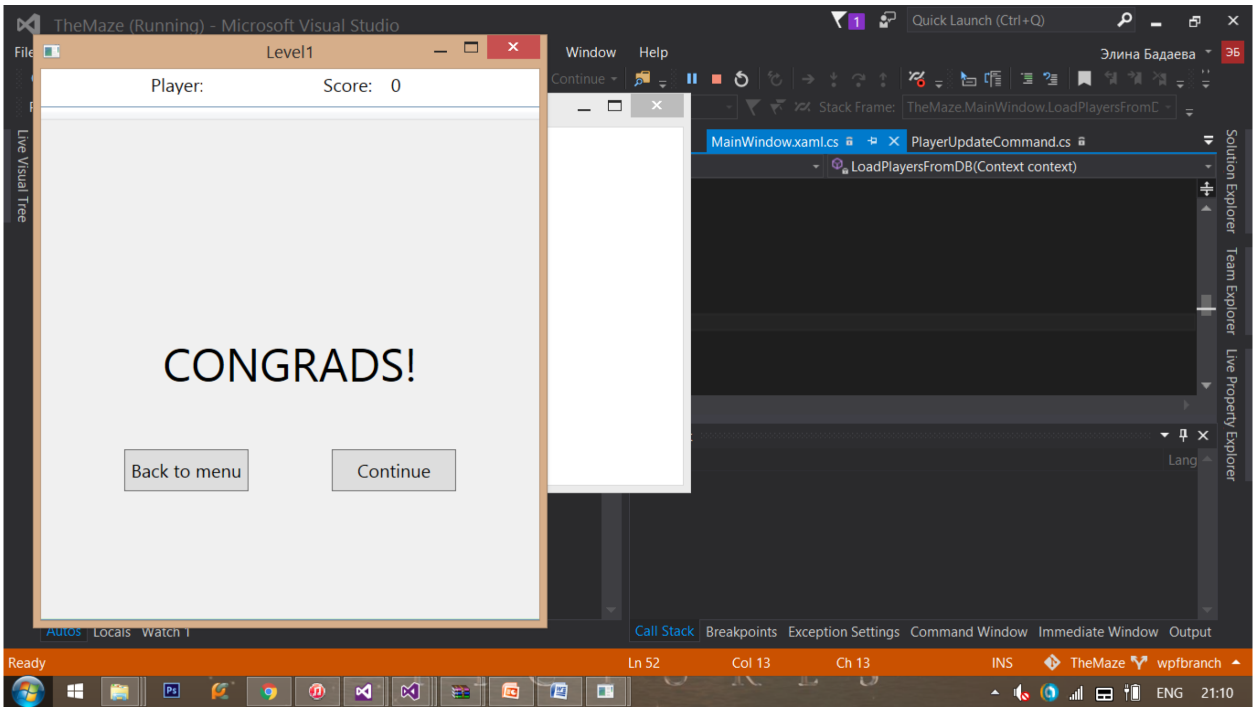
```
class PlayerUpdateCommand : ICom
{
    public PlayerUpdateCommand(P
    {
        _ViewModel = viewModel;
    }

    public PlayerViewModel _View

    public bool CanExecute(objec
    {
        return _ViewModel.CanRe
```

A breakpoint could not be inserted at this location.

If you win, you will get congratulations and an opportunity to go back to menu or to continue the game.


TheMaze (Running) - Microsoft Visual Studio
Level1
Player:
Score: 0
CONGRADS!
Back to menu
Continue
MainWindow.xaml.cs
PlayerUpdateCommand.cs
LoadPlayersFromDB(Context context)
Stack Frame: TheMaze.MainWindow.LoadPlayersFromD
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output
Ready
Ln 52 Col 13 Ch 13 INS
ENG 21:10