МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут **КНІТ**
Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 2
З дисципліни: *"Конструювання програмного забезпечення "*
На тему: *"Лабораторна WPF"*

Лектор:
доц. каф. ПЗ
Сердюк П. В.

Виконав:
ст. гр. ПЗ-35
Хруставчук М.Л.

Прийняв:
асистент каф. ПЗ
Яценко Р. О.

« ____ » _____ 2024 р.

∑= _____ .....

Львів – 2024

Тема роботи: Лабораторна WPF.

Мета роботи: Навчитися працювати із WPF та реалізувати архітектурний патерн MVVM.

ЗАВДАННЯ

Реалізувати основні форми графічного інтерфейсу відповідно до проекту, наприклад: сплеш-скрін, головне меню, контекстне меню, меню закінчення рівня, і т.д.

Варіант проєкту:

Риболовля, збір снастей, вибір локацій, різні сценарії - вибір персонажа з характеристиками (швидкість, уважність, кмітливість, тощо).

ХІД ВИКОНАННЯ

1. Код розробленої програми

Повністю проєкт можна подивитися тут

RelayCommand.cs

```
using System.Windows.Input;

public class RelayCommand : ICommand
{
    private readonly Action<object> execute;
    private readonly Predicate<object> canExecute;

    public RelayCommand(Action<object> execute, Predicate<object> canExecute =
null)
    {
        this.execute = execute;
        this.canExecute = canExecute;
    }

    public event EventHandler CanExecuteChanged
    {
        add { CommandManager.RequerySuggested += value; }
        remove { CommandManager.RequerySuggested -= value; }
    }

    public bool CanExecute(object parameter)
    {
        return canExecute == null || canExecute(parameter);
    }
```

```
    public void Execute(object parameter)
    {
        execute(parameter);
    }
}
```

StartViewModel.cs

```
using System.Windows.Input;
using System.Windows;

namespace FishingGame.ViewModel
{
    public class EndViewModel : ViewModelBase
    {
        public ICommand CloseCommand { get; }
        public EndViewModel()
        {
            CloseCommand = new RelayCommand(CloseWindow);
        }
        private void CloseWindow(object parameter)
        {
            if (parameter is Window window)
            {
                window.Close();
            }
        }
    }
}
```

EndViewModel.cs

```
using System.Windows.Input;
using System.Windows;

namespace FishingGame.ViewModel
{
    public class EndViewModel : ViewModelBase
    {
        public ICommand CloseCommand { get; }
        public EndViewModel()
        {
            CloseCommand = new RelayCommand(CloseWindow);
        }
        private void CloseWindow(object parameter)
        {
            if (parameter is Window window)
            {
                window.Close();
            }
        }
    }
}
```

ViewModelBase.cs

```
using System.ComponentModel;

namespace FishingGame.ViewModel
{
    public class ViewModelBase : INotifyPropertyChanged
    {
        public event PropertyChangedEventHandler PropertyChanged;
        protected void OnPropertyChanged(string propertyName)
        {
```

```
            PropertyChanged?.Invoke(this, new
PropertyChangedEventArgs(propertyName));
        }
    }
}
```

2. Результати виконання розробленої програми

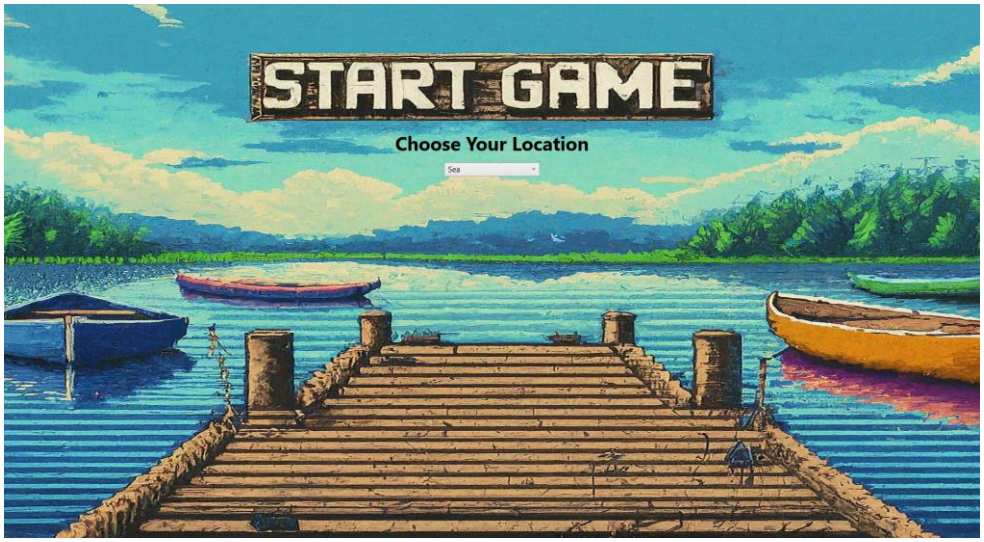

Рис. 1. Початкове меню гри

Рис. 2. Основне меню гри

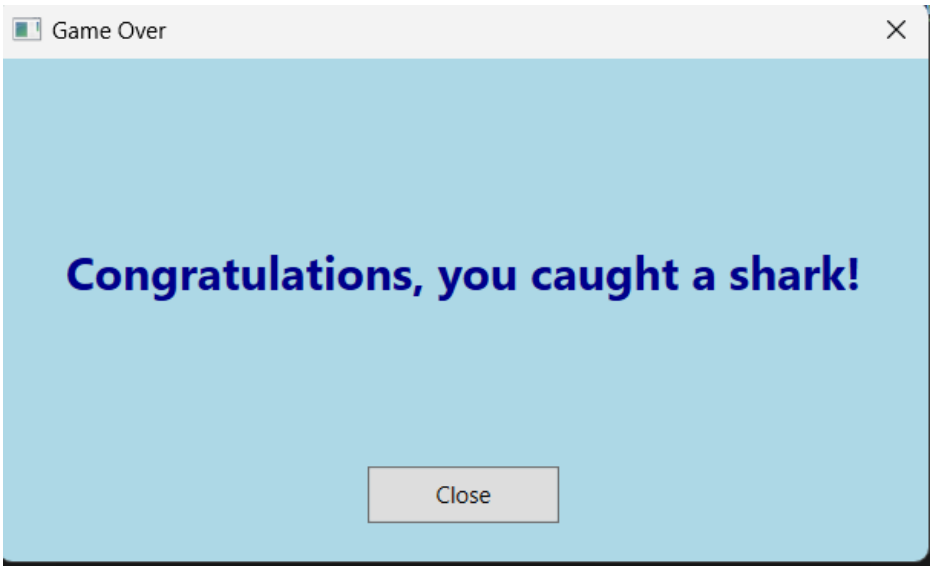

Рис. 3. Фінальне вікно гри

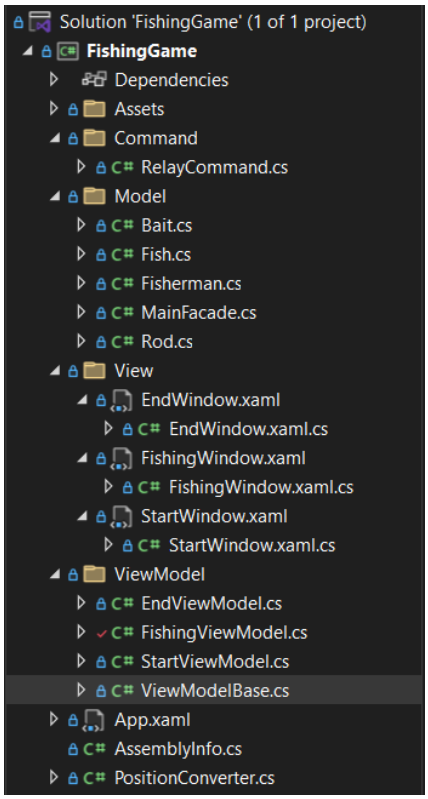

Рис. 4. Файлова архітектура проєкту

ВИСНОВКИ

У цій лабораторній роботі я навчився реалізовувати шаблон проектування MVVM (Model-View-ViewModel) у додатку на базі WPF (Windows Presentation

Foundation). Цей шаблон дозволяє відокремити логіку, інтерфейс користувача та дані, що підвищує підтримуваність і тестованість коду.

Також навчився розділенню відповідальностей за допомогою MVVM, а саме зрозумів, як Model, View і ViewModel виконують різні функції в додатку, забезпечуючи незалежність кожного компонента. Це полегшує внесення змін у додаток і дозволяє тестувати кожен компонент окремо.