**Правительство Российской Федерации**

**Государственное образовательное бюджетное учреждение**

**высшего профессионального образования**

**«Национальный исследовательский университет**

**Высшая школа экономики»**

**Факультет бизнеса и менеджмента**

**Школа бизнес-информатики**

**Кафедра управления информационными системами и цифровой инфраструктурой**

**Пояснительная записка к проекту “Puzzle Game”**

**по дисциплине «Программирование»**

**Выполнили**

**Студенты группы № ББИ142**

**Репецкая Т.О.**

**Рязанов И.В.**

**Позднышева С.В.**

**Руководитель**

**старший преподаватель, к.т.н.**

**С.Г. Ефремов**

**Москва 2015**

**Название программы:** Puzzle Game

**Аннотация:** Программа представляет из себя игру, в которой игроку (пользователю) нужно собрать картинку из кусочков (пазл). Пользователь может выбрать уровень сложности и один из 2 режимов игры, при этом он имеет возможность выбрать картинку из базы данных, либо загрузить свою.

**Адрес удаленного репозитория:** <https://github.com/TanyaFox/ProjectPuzzleGame.git>

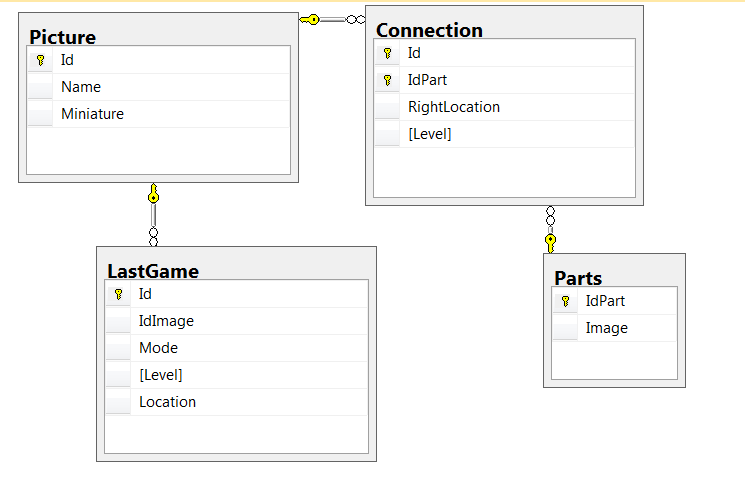
**Распределение ролей:**

Позднышева Софья: Моделирование и разработка базы данных, а также создание методов взаимодействия программы с БД, отладка программы.

Рязанов Игорь: Создание основной логики приложения, необходимой для осуществления игры, структура приложения, отладка программы.

Репецкая Татьяна: Создание графического интерфейса (GUI) на базе WPF, а также связующих компонентов для логики и интерфейса, отладка программы.

**Диаграмма базы данных:**



**Описание классов программы:**

**Классы баз данных**

**Класс DataBase.cs**

Содержит методы для взаимодействия с базой данных SqlServer.

Метод LoadMiniatures возвращает миниатюры всех картинок, содержащихся в базе данных в виде переменных типа Miniature. Он запрашивает информацию в базе данных при помощи sql-запроса.

Метод LoadPuzzle служит для загрузки пазла (кусочков изображения), соответствующего определенной сложности игры. Метод вызывает хранимую процедуру базы данных «ЗагрузкаПазла», считывает поток данных и возвращает словарь типа Расположение картинки-Картинка.

Метод SaveGame позволяет сохранить в базе данных параметры текущей игры: картинку, уровень игры, режим, расположение частей пазла. Метод вызывает хранимую процедуру базы данных "СохранениеИгры" и передает ей все перечисленные параметры. Метод ничего не возвращает.

Метод AddPicture позволяет добавить новую картинку в базу данных. Метод вызывает хранимую процедуру базы данных «ДобавлениеИзображения», а также запрашивает идентификатор добавленной картинки при помощи sql-запроса. Метод возвращает идентификатор добавленной картинки.

Метод AddPartsOfPicture позволяет добавить часть пазла и информацию о её правильном местоположении. Метод вызывает хранимую процедуру базы данных "ДобавлениеЧасти" и передает информацию о части пазла и всем изображении(идентификатор всего изображения, картинка части пазла, уровень сложности, которому она соответствует и правильное расположение этой части пазла). Метод ничего не возвращает.

Метод LoadGame позволяет загрузить информацию о последней запущенной игре: Идентификатор изображения, уровень сложности, тип игры и текущее расположение кусочков пазла. Метод обращается к базе данных при помощи sql-запроса. Метод возвращает переменную типа Game.

**Интерфейсы**

**Интерфейс IField.cs**

Объединяет поля для 2 типов игры. Имеет два свойства (ListCell и ImagePieces), а также общий метод CellChange. Подробнее о них в описании классов, наследующих интерфейс.

**Интерфейс INavigationService.cs**

Задает вид класса для перехода по окнам, имеет определение метода NavigateTo(string windowName).

**Интерфейс INavigationServiceGames.cs**

Задает вид класса для перехода по окнам игры, а именно, здесь находятся поля игр разных уровней и режимов, имеет определение метода NavigateTo(string windowName, int id, int level).

**Классы Model**

**Класс Miniature.cs**

Содержит три поля: IdImage (идентификатор изображения), ImageName (Название изображения), Picture (Маленькое изображение картинки) и конструктор.

**Класс Cell.cs**

Класс представляет собой модель одного участка поля. Он содержит следующие поля: IsNotCorrect (bool, показывает, находится ли в данной ячейке неправильная деталь), CurrentElement (int, истинный номер детали, находящейся в данной ячейке), Image (byte[], кусок картинки в данной ячейке).

**Класс PuzzleMethods.cs**

Класс содержит в себе основные методы для работы с игры. Метод LoadSave создаёт игру по сохранённым данным. Метод CreateNewGame генерирует новую игру. Методы DefineGameModes и DefineDifficultyLevels дают доступ к словарям со списком режимов и сложностей игры соответственно. Метод FormMode образует идентификатор для окна игры.

InitiateFragmentation запускает нарезку картинки и загрузку в БД фрагментов для всех режимов. SendFragments нарезает изображение и отправляет кусочки по одному в БД. BitmapToImageSource - это конвертер изображения Bitmap в BitmapImage.

**Класс Game.cs**

В этом классе сохраняются данные игры для сохранения. Он имеет поля ImageID (int, номер картинки в БД), Difficulty (int, сложность от 1 до 3), Type (int, режим игры, 1 - пятнашки, 2 - кусочки), PartsLocation (string, номера кусочков, стоящих в ячейках, записаны через запятую).

**Класс Field1.cs**

Класс представляет собой модель поля типа пятнашки. Он содержит поле, состоящее из списка ячеек Cell, а также список кусочков текущей картинки. В классе есть два конструктора (для создания новой игры со случайным положением деталей и для загрузки старой) и метод CellChange, который меняет содержимое двух ячеек местами и проверяет, правильно ли стоит деталь.

**Класс Field2.cs**

Класс представляет собой модель поля типа кусочки. Он содержит поле, состоящее из списка ячеек Cell, а также список кусочков текущей картинки. В классе есть два конструктора (для создания новой игры и для загрузки старой) и метод CellChange, который заполняет ячейку и проверяет, правильно ли встала деталь.

**Классы ViewModel**

**Класс MainWindowViewModel.cs**

Класс является связующим для логики и главного окна MainWindowView**.** В нем находятся команды и соответствующие им обработчики событий для главного окна: ButtonNewGameCommand (ButtonNewGameClick) - обрабатывает нажатие кнопки “Новая игра” и переводит в окно для выбора игры, ButtonSavedGameCommand (ButtonSavedGameClick) - обрабатывает кнопку “Продолжить игру” и загружает последнюю сохраненную игру, ButtonAuthorsCommand (ButtonAuthorsClick) - обрабатывает кнопку “Авторы” и переходит на окно с авторами.

**Класс BeforeNewGameWindowViewModel.cs**

Класс является связующим для логики и окна BeforeNewGameWindowView**.** В нем находятся команды и соответствующие им обработчики событий для этого окна: ButtonFromDBCommand (ButtonFromDBClick) - обрабатывает кнопку “Из базы” и переходит на окно для выбора параметров игры с картинками из базы данных, ButtonUploadCommand (ButtonUploadClick) - обрабатывает нажатие кнопки “Свою” и переходит в окно с параметрами для загрузки картинки и выбора параметров игры в этом случае.

**Класс NewGameWindowViewModel.cs**

Класс является связующим для логики и окна NewGameWindowView**.** В нем находятся команды и соответствующие им обработчики событий для этого окна: ButtonPlayCommand (ButtonPlayClick) - обрабатывают нажатие кнопки “Играть!”. Кроме того, в нем есть свойства: GameMode - словарь режимов игры, LevelDifficulty - словарь уровней сложности, ListOfPictures - список картинок из БД, SelectedImage - выбранная картинка, Mode - выбранный режим, Difficulty - выбранный уровень сложности. Переменные \_navigationServiceGame, pz и db нужны для реализации переходов между окнами, использования методов для игры и использования методов для работы с БД соответственно.

**Класс CustomNewGameWindowViewModel.cs**

Класс является связующим для логики и окна CustomNewGameWindowView**.** В нем находятся команды и соответствующие им обработчики событий для этого окна: ButtonPlayCommand (ButtonPlayClick) - обрабатывают нажатие кнопки “Играть!”, ButtonUploadPictureCommand (ButtonUploadPictureClick) - . Кроме того, в нем есть свойства: GameMode - словарь режимов игры, LevelDifficulty - словарь уровней сложности, ListOfPictures - список картинок из БД, SelectedImage - выбранная картинка, Mode - выбранный режим, Difficulty - выбранный уровень сложности. Переменные \_navigationServiceGame, pz и db нужны для реализации переходов между окнами, использования методов для игры и использования методов для работы с БД соответственно.

**Класс GameWindowViewModel.cs**

Класс является связующим для логики игры в режиме “Пятнашки” и окна GameWindowView**.** В нем находятся команды и соответствующие им обработчики событий для этого окна: ButtonPressedCommand (ButtonPressedClick) - обработчик нажатия на кнопку кусочком картинки, это параметрическая команда, обработчик меняет местами два кусочка в соответствии с правилами, ButtonSavedGameCommand (ButtonSavedGameClick) - обрабатывает нажатие на кнопку “Сохранить игру” и загружает сохраненную игру в БД. Метод CallPopulateMethod(id, level) в параллельном режиме .

Метод PopulateProperties(int id, int level) .

Также в нем есть следующие свойства для связи с GUI: Image - список кусочков изображения, IsEnabled - список состояний доступности кнопок с кусочками этого изображения, и переменные: pz и db, нужные для реализации использования методов для игры и использования методов для работы с БД соответственно.

**Класс GameDragDropModeWindowViewModel.cs**

**Класс Command.cs**

Класс реализует интерфейс IСommand и таким образом дает возможность создавать команды и привязывать к ним обработчики графики.

**Классы View**

**Класс NavigationService.cs**

Реализует интерфейс INavigationService, хранит в себе словарь окон и метод для навигации по ним.

**Класс NavigationServiceForGames.cs**

Реализует интерфейс INavigationServiceGames, хранит в себе словарь окон с полями для игры и метод для навигации по ним.

**Окно/класс MainWindowView.xaml/cs**

См. описание GUI и ViewModels.

**Окно/класс BeforeNewGameWindowView.xaml/cs**

См. описание GUI и ViewModels.

**Окно/класс NewGameWindowView.xaml/cs**

См. описание GUI и ViewModels.

**Окно/класс CustomNewGameWindowView.xaml/cs**

См. описание GUI и ViewModels.

**Окна/классы GameWindowView.xaml/cs, GameMiddleWindowView.xaml/cs и GameHardWindowView.xaml/cs**

См. описание GUI и ViewModels.

**Окна/классы GameDragDropModeWindowView.xaml/cs, GameDragDropModeMiddleWindowView.xaml/cs и GameDragDropModeHardWindowView.xaml/cs**

См. описание GUI и ViewModels.

**Окно/класс AboutAuthorsWindowView.xaml/cs**

См. описание GUI и ViewModels.

**Интерфейс приложения**

Ниже представлены скриншоты экранов приложения с кратким описаниям к ним.

1. Рисунок 1 - MainWindowView.xaml - главное окно приложения. Имеет три кнопки: “Новая игра”, “Продолжить игру” и “Авторы”. Здесь всё интуитивно ясно.
2. Рисунок 2 - BeforeNewGameWindowView.xaml - окно появляется при выборе новой игры на предыдущем шаге. Здесь пользователю предлагается выбрать, какую картинку он хочет собрать, готовую из БД, или свою.
3. Рисунок 3 - NewGameWindowView.xaml - окно, в котором пользователю предлагается выбрать режим игры (Пятнашки: нужно менять кусочки местами или Кусочки: нужно брать кусочки из кучи), а также 1 из 3 уровней сложности (легко, средне и сложно). Тут же пользователь выбирает картинку. По нажатию на кнопку “Играть!” открывается окно с игрой. Победа знаменуется соответствующим сообщением.
4. Рисунок 4 - NewCustomGameWindowView.xaml - окно, в котором пользователю также предлагается выбрать режим игры (Пятнашки: нужно менять кусочки местами или Кусочки: нужно брать кусочки из кучи), а также 1 из 3 уровней сложности (легко, средне и сложно). Но тут уже пользователь выбирает свою картинку формата jpeg/jpg. Это делается нажатием на кнопку “Загрузить”. По нажатию на кнопку “Играть!” открывается окно с игрой. Победа знаменуется соответствующим сообщением.
5. Рисунок 5 - AboutAuthorsWindowView.xaml - это окно, в котором находится информация о создателях приложения.

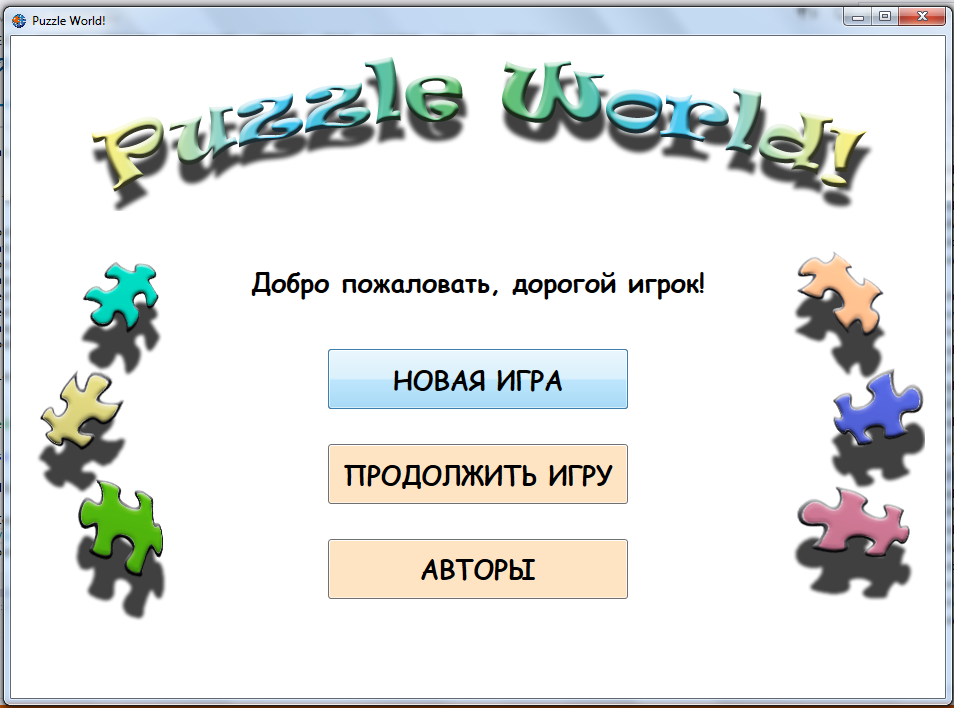


Рис. 1 Главное окно приложения (MainWindowView.xaml)

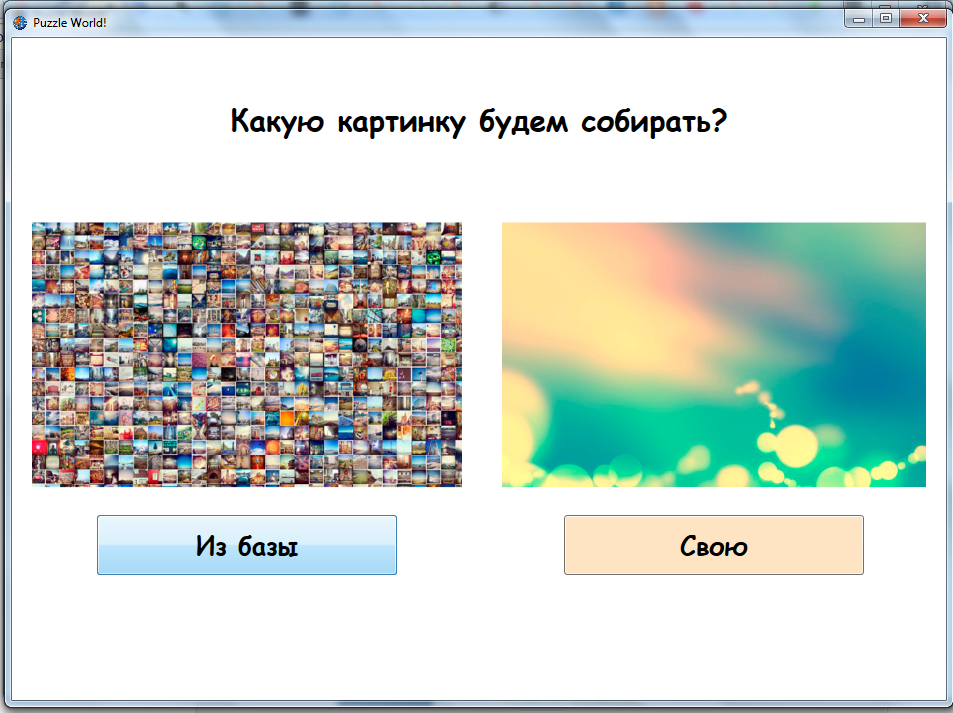


Рис.2 Окно выбора типа картинки (BeforeNewGameWindowView.xaml)

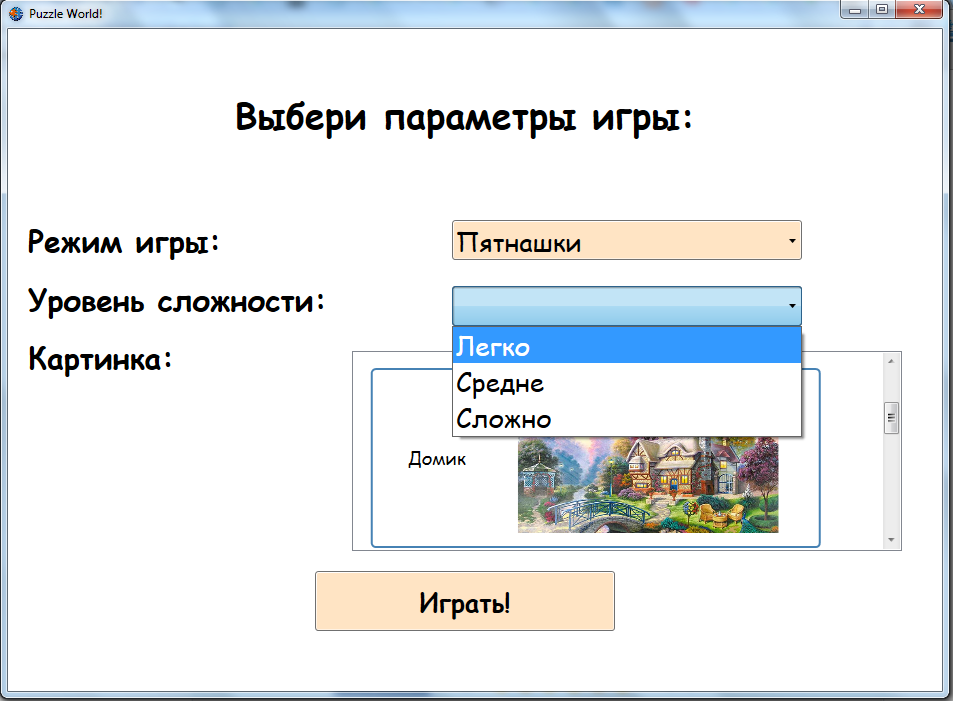


Рис.3 Окно параметров игры при выборе картинки из базы (NewGameWindowView.xaml)

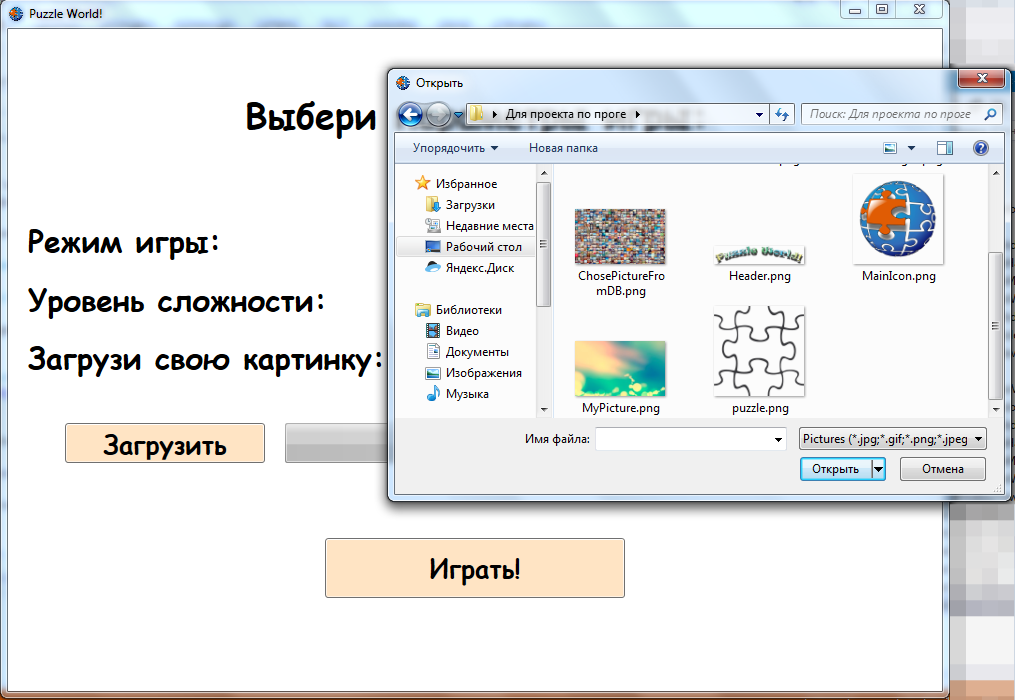


Рис.4 Окно параметров игры при выборе своей картинки (NewCustomGameWindowView.xaml)



Рис.5 Окно с информацией об авторах (AboutAuthorsWindowView.xaml)

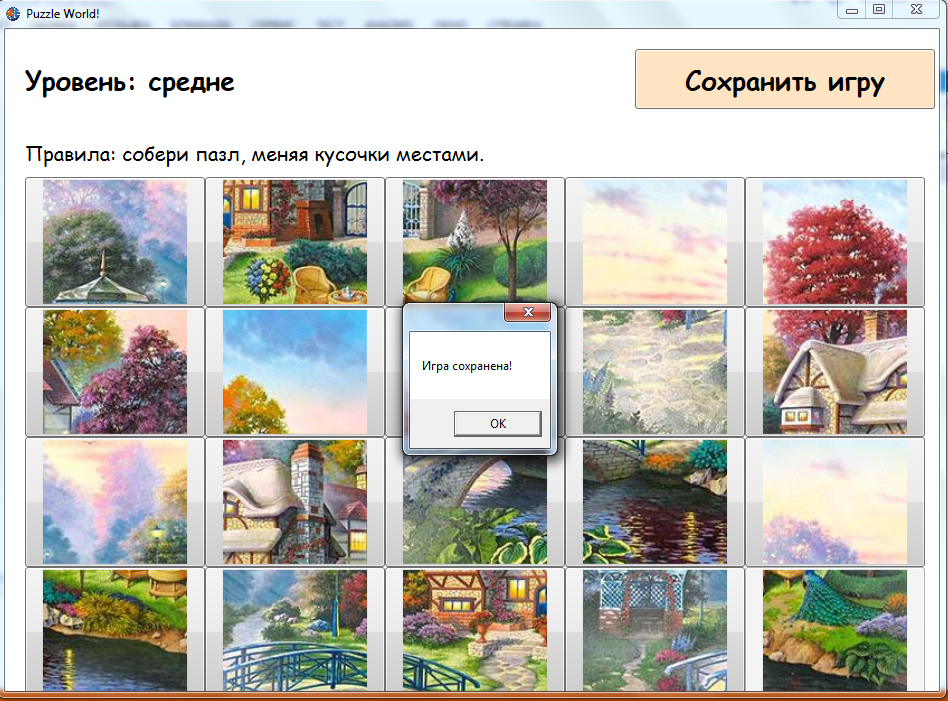


Рис.6 Окно с игрой режим “Пятнашки”