**Пояснительная записка**

* *Название проекта:*

Игра домино (The game of dominoes)

* *Аннотация – краткое описание программы и её функциональности:*

Открыв нашу игру, пользователь видит игровое поле. В левом верхнем углу расположена кнопка «Start game» (меняется на «Restart game», сразу после начала игры, чтобы начать игру заново), при нажатии на которую пользователю предлагают вписать свой никнейм, после чего начинается игра; в правом верхнем углу расположена кнопка «Pick new tile» - пока не началась игра, она неактивна – эта кнопка нужна для того, чтобы, если у игрока нет нужных костей для хода, он мог брать дополнительные кости из «базара» (при нажатии на эту кнопку) до тех пор, пока у него не появится нужной для игры кости; посередине между этими двумя кнопками пользователь, начав игру, видит приветствие «Hello, \*\*\*!», где \*\*\* - это никнейм пользователя. Ниже расположены ещё 2 поля: «AI tiles count:» и «Tiles in the stock:». Первый из них показывает общее количество костей у противника (у компьютера, AI – Artificial Intelligence), а второй – количество костей, доступных в «базаре».

Правила просты: игрокам раздаётся по 7 костей, остальные размещаются в «базаре». Начинает игрок, у которого на руках находится «дубль ноль» (0-0) или любой другой минимальный дубль, если нет (0-0) (если же нет ни одного дубля, компьютер ищет дубль в «базаре»). Допустим, у первого игрока есть «дубль ноль» (0-0), и он ходит первым. Второй игрок должен сделать ход той костью, у которой есть 0 на одной из сторон – например, (0-1), (0-2) и т.д. Если нет подходящей для ходя кости, игрок обращается к «базару» - кнопка «Pick new tile». Игра заканчивается, когда один из игроков выложит последнюю кость.

* *Центральный адрес хранилища:*

[https://github.com/smitford/Dominoes.git](http://vk.com/away.php?to=https%3A%2F%2Fgithub.com%2Fsmitford%2FDominoes.git" \t "_blank)

* *Члены команды и их роли:*

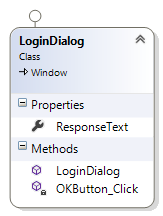
1. Калмыков Андрей – логика
2. Хорева Любовь – графический интерфейс пользователя
3. Мусорина Наталья – база данных, тесты

* *Список классов с кратким описанием каждого из них:*

**GUI**

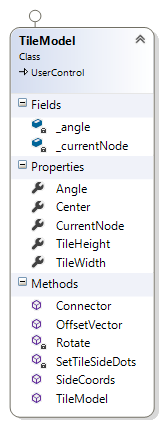
**LoginDialog**

Окно, в котором пользователя просят ввести свой псевдоним.



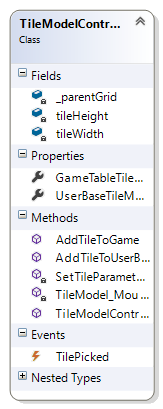
**TileModel**

UI элемент доминошки. Содержит информацию об узле (доминошке), текущих координатах центра и угле наклона. Connector – ищет координаты стороны доминошки относительно указанной стороны света. OffsetVector – вектор из центра доминошки в указанную сторону доминошки. SideCoords – координаты центра + OffsetVector.



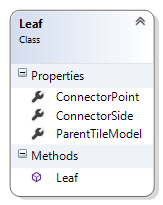
**TileModelController**

Класс-контроллер содержит методы для управления моделями доминошек в интерфейсе. Содержит событие, срабатывающее при выборе доминошки из своей «колоды».



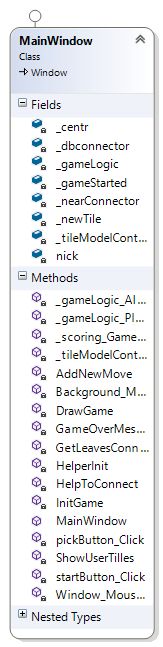
**Leaf**

Описывает свободные точки, к которым можно присоединить доминошки.



**MainWindow**

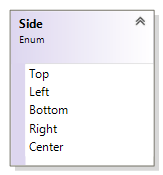
Главное окно игры. Содержит кнопки начала игры и подбора новой доминошки. Рассчитывает координаты доминошек и обрабатывает все события.



**AI  
Side**

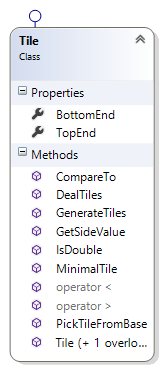
Перечисление сторон доминошки.

          ---0---  
         | \*    |  
         |    \* |  
        |2|-----|3|  
         | \*   \* |  
         | \*   \* |  
          ---1---  
      
Top = 0,  
Bottom = 1,  
Left = 2,  
Rigt = 3,  
Center = 4



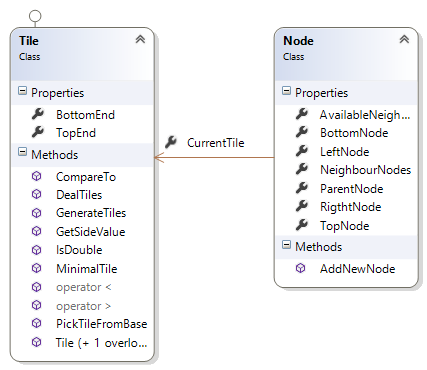
**Tile**

Класс, содержащий информацию о количестве точек в верхней и нижней части, а также методы для генерирования новых доминошек, ранжирования, раздачи и прочее.



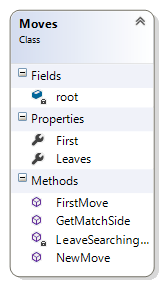
**Node**

Класс ассоциируется с классом Tile. Содержит свойство CurrentTile и ссылки на соседние доминошки. Представляет из себя узел связного списка. Содержит метод добавления нового соседнего элемента.



**Moves**

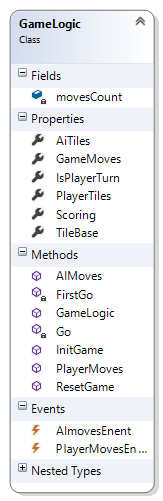
Оболочка связного списка. Содержит метод добавления нового элемента к списку и рекурсивный метод нахождения листьев дерева (узлов со свободными связями).



**GameLogic**

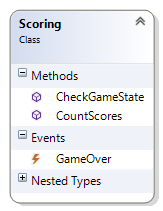
Класс включает такие методы, поддерживающие ход игры, как:

* InitGame – инициализация игры. Генерация доминошек и раздача (в списки PlayerTiles, AiTiles, BaseTiles). Определения игрока, который ходит первым
* Go – метод принимающий данные о новом ходе игрока/компьютера для добавления этих данных в список и передачи права хода другому игроку
* AIMoves – метод, описывающий логику принятия решения компьютера. Выбирает первую подходящую доминошку и ставит на стол. Если подходящих доминошек нет, то подбирает из базара (от английского stock)
* PlayerMoves – метод, вызываемый извне, при ходе игрока.
* AIMovesEvent, PlayerMovesEvent – события, срабатывающие при решении компьютера об автоматическом ходе компьютером или игроком



**Scoring**

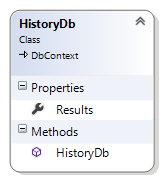
Содержит метод проверки состояния игры, который принимает текущие параметры хода игры, и если условия удовлетворяют правилам окончания игры, то подсчитывает очки и вызывает событие GameOver



**DB**

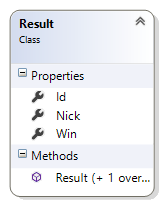
**HistoryDb**

База данных на основе EntityFramework CodeFirst. Содержит класс (таблицу) Result



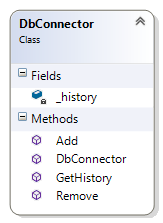
**Result**

Класс (таблица) содержащий параметры исхода игры: id партии, nickname игрока и булевую переменную, которая говорит о том, выиграл игрок или проиграл.



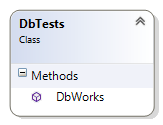
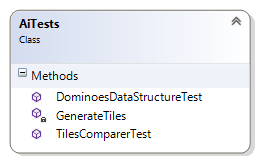
**DbConnector**

Контроллер, инициализирующий соединение с БД и хранящий методы добавления и удаления новой записи в таблицу Results, а также метод, возвращающий всю таблицу



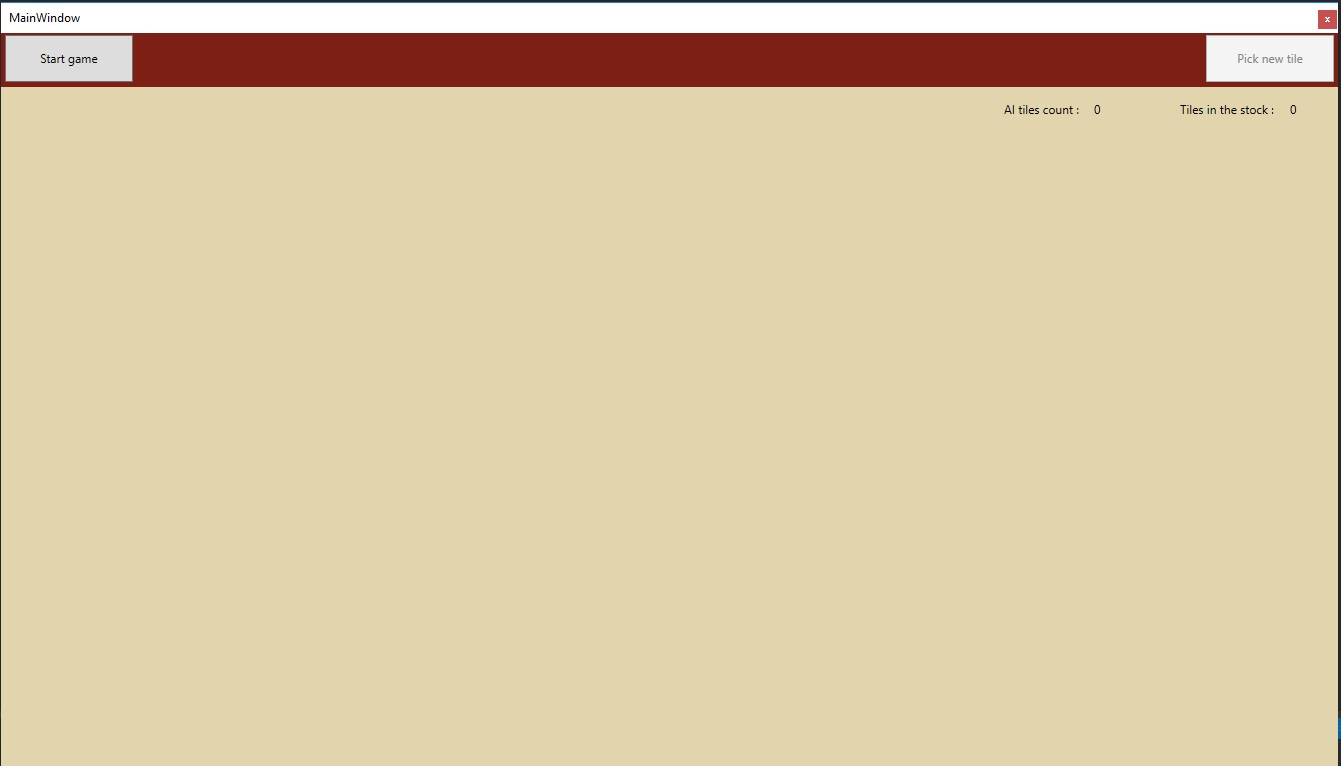
**Tests**

* Тест проверяющий правильность соединений связного списка
* Тест правильности работы компаратора(сравнения) доминошек
* Тест правильности работы БД

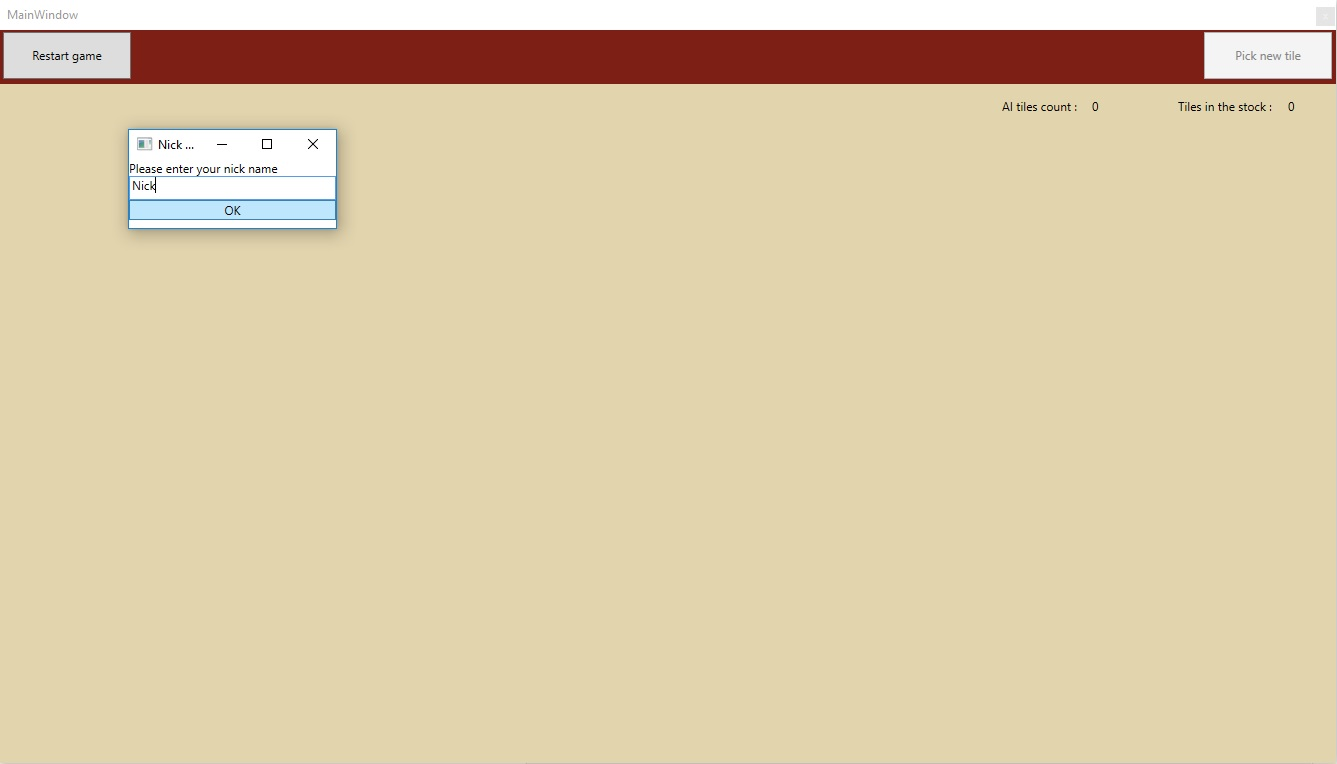


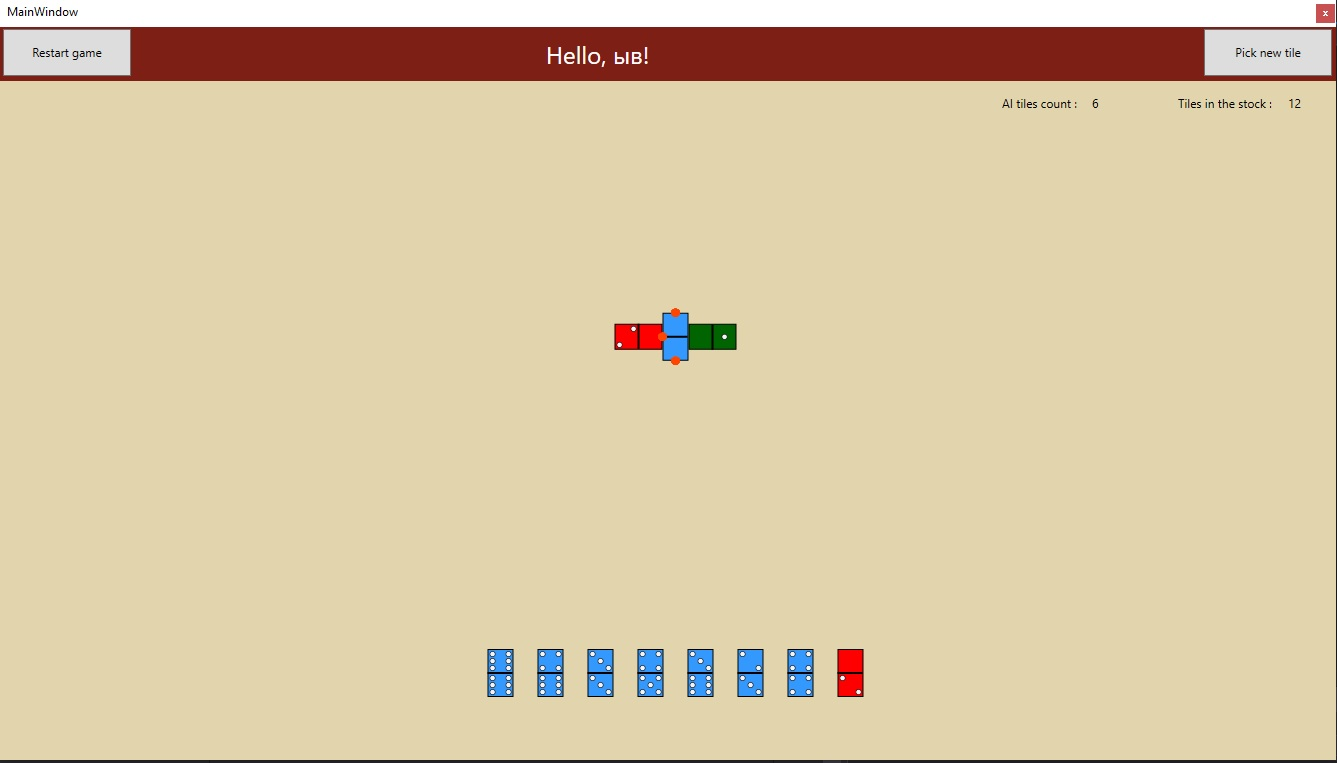
* *Интерфейс программы:*

Первый экран, который видит пользователь при запуске нашей игры – чистое игровое поле:

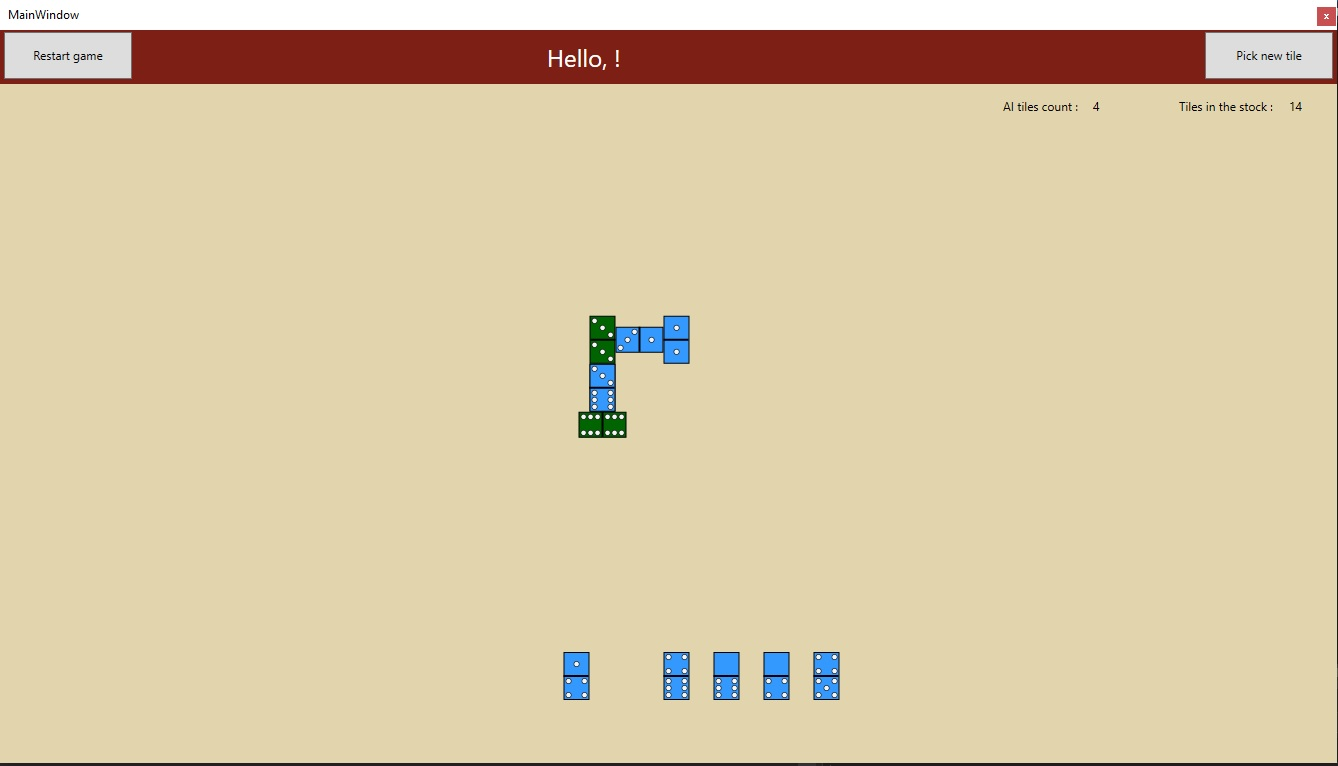


Экран ввода никнейма пользователем для начала игры:



Экран выбора кости:  


Экран хода:



Экран конца игры:

