D&D Helper

Краткое описание.

Приложение-помощник по настольной ролевой игровой системе Dungeons and Dragons. Предоставляет функционал как для игроков, так и для мастеров. Обладает интуитивным современным дизайном, созданном с помощью Windows Forms. Само приложение написано на языке C#. Материалы взяты из открытых источников.

Общий функционал программы.

База предметов

База монстров

База предметов

База монстров

Создание своего контента

База персонажей

Книга заклинаний

Журнал приключений

Расчет инициативы

Бросок кубиков

База персонажей

Создание персонажей

Создание персонажей

Книга заклинаний

Бросок кубиков

База персонажей

D&D

HELPER

Схема классов.

Utilities

*(Функционал*

*Выход из программы;)*

CommonChoose

*(Выбор функционала:*

*Игрок или Мастер;*

*Выход из программы)*

ItemsBase

*(База предметов)*

MonstersBase

*(База монстров)*

InitiativeCounter

*(Расчет инициативы)*

AdventuresJournal

*(Журнал приключений)*

OwnContentBuilder

*(Создание собственного контента)*

DiceRoller

*(Бросок кубиков)*

SpellsBase

*(База заклинаний)*

CharactersBase

*(База персонажей)*

CharactersCreator *(Создание персонажей)*

IUserInterface

*(Функционал мастера;*

*Выход из программы)*

*IUserInterface (Функционал игрока;*

*Выход из программы)*

**Не готово**

**В работе**

**Готово**

CommonChoose class

**Код:**

class CommonChoose

{

public static int UserChoice { get; private set; } = 0;

public static void UserIsPlayerChoice()

{

UserChoice = 1;

}

public static void UserIsMasterChoice()

{

UserChoice = 2;

}

}

**Методы**:

***UserIsPlayerChoice****()*

*// UserChoice = 1*

***UserIsMasterChoice****()*

*// UserChoice = 2*

**Переменные:**

*public int UserChoice*

*// определяет кто пользователь – игрок (1) или мастер (2)*

IUserInterface class

***Методы****:*

*public* ***CreateCharacter****()*

*// создать персонажа*

*public* ***OpenSpellsBase****()*

*// открыть базу заклинаний*

*public* ***OpenCharactersBase****()*

*// открыть базу персонажей*

*public* ***OpenItemsBase****()*

*// открыть базу предметов*

*public* ***OpenDiceRoller****()*

*// открыть меню для броска кубиков*

*public* ***OpenMonstersBase****()*

*// открыть базу монстров*

*public* ***OpenInitiativeCounter****()*

*// открыть счетчик инициативы*

*public* ***OpenAdventuresJournal****()*

*// открыть Журнал Приключений*

*public* ***OpenOwnContentBuilder****()*

*// открывает меню создания собственного контента*

**Переменные:**

не нужны

**Код:**

interface IUserInterface

{

void CreateCharacter();

void OpenSpellsBase();

void OpenCharactersBase();

void OpenItemsBase();

void OpenDiceRoller();

void OpenMonstersBase();

void OpenInitiativeCounter();

void OpenAdventuresJournal();

void OpenOwnContentBuilder();

}

Utilities class

**Переменные:**

не нужны

***Методы****:*

*public* ***CreateCharacter****() : IPlayerInterface, IMasterInterface*

*// создать персонажа – CharactersCreator.CreateCharacter()*

*public* ***OpenSpellsBase****() : IPlayerInterface, IMasterInterface*

*// открыть базу заклинаний – SpellsBase.OpenSpellBase()*

*public* ***OpenCharactersBase****() : IPlayerInterface, IMasterInterface*

*// открыть базу персонажей – CharactersBase.OpenCharacterBase()*

*public* ***OpenItemsBase****() : IPlayerInterface, IMasterInterface*

*// открыть базу предметов – ItemsBase.OpenItemsBase*

*public* ***OpenDiceRoller****() : IPlayerInterface, IMasterInterface*

*// открыть меню для броска кубиков – DiceRoller.OpenDiceRoller*

*public* ***OpenMonstersBase****() : IPlayerInterface, IMasterInterface*

*// открыть базу монстров – MonstersBase.OpenMonsterBase()*

*public* ***OpenInitiativeCounter****() : IMasterInterface*

*// открыть счетчик инициативы – InitiativeCounter.OpenInitiaiveCounter()*

*public* ***OpenAdventuresJournal****() : IMasterInterface*

*// открыть Журнал Приключений – AdvenduresJournal.OpenAdventureJournal()*

*public* ***OpenOwnContentBuilder****() : IMasterInterface*

*// открывает меню создания собственного контента – OwnContentBuilder.OpenOwnContentBuilder()*

**Код:**

class Utilities : IUserInterface

{

public void CreateCharacter()

{

}

public void OpenSpellsBase()

{

}

public void OpenCharactersBase()

{

}

public void OpenItemsBase()

{

}

public void OpenDiceRoller()

{

}

public void OpenMonstersBase()

{

}

public void OpenInitiativeCounter()

{

}

public void OpenAdventuresJournal()

{

}

public void OpenOwnContentBuilder()

{

}

}

DiceRoller class

**Код:**

class DiceRoller

{

private static Random rnd = new Random();

public static int DiceChoice { get; set; }

public static void OpenDiceRoller()

{

}

private void DiceRoll()

{

rnd.Next(1, DiceChoice);

}

}

**Переменные:**

*private static Random* ***rnd*** *= new Random()*

// рандомное число для броска кубиков

*public static int* ***DiceChoice*** *{get; set}*

// количество граней

***Методы****:*

*public static void* ***OpenDiceRoller****()*

*// визуал для утилиты*

*private void* ***DiceRoll****()*

*// бросок кубика*

CharactersCreator class

**Код:**

***Методы****:*

*public static void* ***OpenCharactersCreator****()*

*// визуал для утилиты*

*private void* ***CreateCharacter****()*

*// создать персонажа*

**Переменные:**

*private static Excel.Application excelCharactersBase = new Excel.Application()*

*// экземпляр приложения*

*private static Excel.Workbook workBookCharactersBase*

*// книга для приложения*

*private static Excel.Worksheet workSheetCharactersBase*

*// страница для приложения*