**В общем и целом**

Проект позволяет создавать свои сценарии, проходить их.

**Характеристики**

Характеристики – набор из шести вещественных чисел.

* В игре можно проиграть только одним образом, допустить выход одной их характеристик из диапазона.
* После выбора действия каждого события все характеристики меняют свое значения.
* Игрок видит все шесть характеристик во все моменты игры

**Карточка**

Карточка (или событие) - основная игровая единица, состоящая из названия события, события, двух вариантов действия на событие, двух последствий на выбранные действия, изменений характеристик, необходимых характеристик, необходимых открытых карточек, карточек, которые откроются после выбора первого или второго действия и картинки

Пример:

* Название: Эпидемия
* Текст события: В Лондоне началась страшная эпидемия холеры, что предпримете?
* Первый вариант действия: Предложить жителям помолиться богу
* Последствие на выбор первого варианта: Бесконечные смерти продолжаются, жители негодуют
* Изменения характеристик, при выборе первого варианта: Популярность: - 40, Коррупция: -5, Инфляция: -15, Армия: -15, Экономика: -20, Бюджет: +20
* Открывающееся карточки, при выборе первого варианта действия: (таких карточек нет)
* Второй вариант действия: Создать комиссию для расследования причин эпидемии
* Последствие на выбор второго варианта: Ждем результатов расследования
* Изменения характеристик, при выборе второго варианта: Популярность: 0, Коррупция: -5, Инфляция: -15, Армия: -15, Экономика: -20, Бюджет: -5
* Открывающееся карточки, при выборе первого варианта действия: Канализация
* Необходимые значения характеристик для появления карточки: Популярность: (нет условия), Коррупция: (нет условия), Инфляция: (нет условия), Армия: (нет условия), Экономика: (нет условия), Бюджет: (нет условия)
* Необходимые открытые ранее карточки для открытия этой карточки: (таких карточек нет)
* Картинка, иллюстрация к событию

**Сценарий**

Сценарий – это набор карточек

Если более развернуто, то сценарий – это одна история, которая рассказывается с помощью событий (карточек). Все карточки после разыгрывания открывают другие карточки (какие карточки, зависит от выбора игрока), соответственно весь сценарий можно представить, как структуру, напоминающую граф, где вершины – события (карточки), из которых исходят ребра двух разных цветов. При этом игрок, проходя сценарий, движется по этому “графу”, идя от одних событий к другим.

Пример:

Пусть в сценарии всего четыре карточки:

1. A, начале открыта, при выборе первого варианта открывается карточка B, а при выборе второго – карточки C и D
2. B, в начале закрыта
3. C, в начале закрыта
4. D, в начале закрыта

Тогда путь игрока может иметь такие виды:

1. Показывается событие A -> игрок выбирает первый вариант действия -> открывается событие B -> показывается событие B -> игра заканчивается
2. Показывается событие A -> игрок выбирает второй вариант действия -> открываются события C и D -> показывается либо событие C, либо событие D -> показывается оставшиеся событие из C и D (которое не показывалось на предыдущем шаге) -> игра заканчивается

**Хранение данных**

Для каждого сценария создается отдельная папка, где будет храниться все, что относится к этому сценарию. В ней хранится 3 элемента, но редактор может добавлять новые:

1. Файл, называющийся “information.txt” и хранящий в себе в себе на отдельных строчках следующие данные:
   1. Название сценария в первой строчке
   2. Названия характеристик во второй строчке через символ “|”
   3. Начальные значения характеристик в третьей строчке через символ “|”
   4. Текст вступления, в файле четвертой строчкой идет символ “<”, далее в n строчек текст вступления, затем в отдельной строчке символ “>”
   5. Название файла, содержащего картинку для вступления
2. Базу данных, называющееся “DB.db” и содержащую три таблицы:
   1. Таблицу для хранения карточек с названием “cards”
   2. Таблицу для хранения карточек, которые открываются после открытия других карточек, с названием “descents”
   3. Таблицу для хранения шести вариантов проигрыша, по одному на каждую характеристику
3. Папка, называющаяся “pictures”. В ней предлагается хранить все картинки, участвующие в данном сценарии

**Процесс одной игры**

1. Из доступных карточек сценария (которые не были выбраны ранее и которые могут быть выбраны) случайным образом выбирается одна
2. Игроку показывается текст события, два варианта действий и картинка, программа ожидает выбор игрока
3. Когда игрок сделает выбор между двумя реакциями на событие, показывается текст последствия на выбранное действие и изменяются значения характеристик
4. Проверяется, все ли характеристики находятся в допустимом диапазоне
   1. Если нет, то игра заканчивается
   2. Если да, то делаем доступными все карточки, которые должны открыться при выборе того или иного действия
5. Цикл начинается сначала

**Класс Main**

Класс “Main” является классом связкой между остальными, хранит в себе информацию о том, какой сценарий открыт, позволяет открывать и закрывать другие классы и окна

**Класс Start**

Класс “Start” открывает начальное окно, позволяет открывать, создавать и удалять сценарии. На открывшемся окне находятся четыре кнопки:

1. Кнопа открытия сценария
2. Кнопка создания сценария
3. Кнопка удаления сценария
4. Кнопка выхода

**Класс Interface**

Класс “Interface” позволяет начать игру, открыть окно для редактирования сценария. Окно, которое открывает данный класс, содержит три кнопки:

1. Кнопка начала игры
2. Кнопка редактирования сценария
3. Кнопка выхода

**Класс Redactor**

Класс “Redactor” показывает таблицы cards, descent, finals, позволяет добавить новую кнопку, изменить карточку, удалить карточку, изменить сценарий, удалить запись из таблицы descent, содержит шесть кнопок:

1. Кнопка создания карточки
2. Кнопка изменения карточки
3. Кнопка удаление карточки
4. Кнопка удаления “наследников”
5. Кнопка редактирования сценария
6. Кнопка выхода

**Класс CreateCard**

Класс “CreateCard“ позволяет создавать и редактировать карточки, содержит множество полей для ввода всех данных, нужных для одной карточки данных, не позволяет сохранять изменения при некорректном вводе данных

**Класс RedactorScenario**

Класс “RedactorScenario” позволяет создавать и редактировать сценарии, содержит множество полей для ввода всех данных, нужных для одного сценария, не позволяет сохранять изменения при некорректном вводе данных

**Класс OneGame**

Класс “OneGame” открывает и управляет окном для игры, содержит 19 кнопок:

1. 6 названий характеристик и 6 значений характеристик
2. Два поля, куда выводятся варианты действий
3. Поле для картинки
4. Поле, куда выводится текст события
5. Две кнопки для выбора действия
6. Кнопка выхода

**Используемые технологии**

Проект писался на языке программирования Python с использованием внешних модулей sys, os, random, shutil, PyQt, sqlite3, также проект взаимодействует с базами данных и с txt файлами