

Содержание

1. Цели работы – стр. 2
2. Характеристика жанра и особенности игры – стр. 3
3. Снимки интерфейса – стр. 4
4. Техническое описание – стр. 6
5. Выводы – стр. 9

Цели работы

1. Использовать приобретенные знания о языке C++ на практике.
2. Изучить библиотеку Qt.
3. Освоить концепцию сигналов и слотов, реализующую событийно-ориентированную парадигму программирования в библиотеке Qt.
4. Написать собственный текстовый квест.

Характеристика жанра и особенности игры

Текстовый квест является поджанром Interactive Fiction (Интерактивная Художественная Литература). Первые игры этого жанра начали создавать практически сразу с появлением первых компьютерных игр.

Основная концепция текстового квеста – взаимодействие с игроком посредством текстовой информации. Игрок читает историю, а в нужные моменты ему предлагается совершить сюжетные выборы.

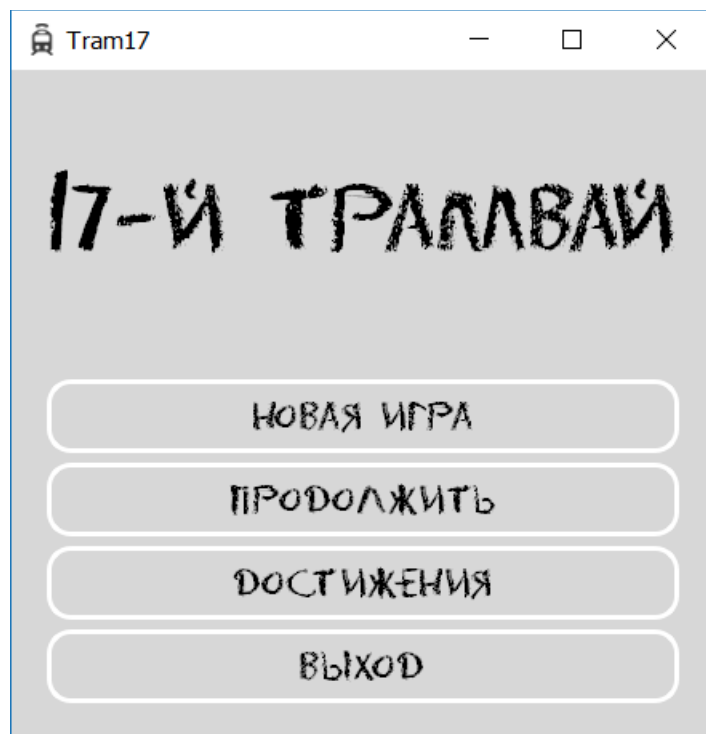
Некоторые квесты написаны на собственных движках, некоторые используют так называемые системы разработки. Самые известные из них: URQ, AXMA Story Maker, QSP, INSTEAD. Многие из систем разработки имеют собственные сообщества и библиотеки, куда любой желающий может добавить своё творение.

Также в некоторых играх другого жанра встречаются встроенные текстовые квесты, внося разнообразие в геймплей. Примером могут служить две части «Космических рейнджеров» от российских разработчиков.

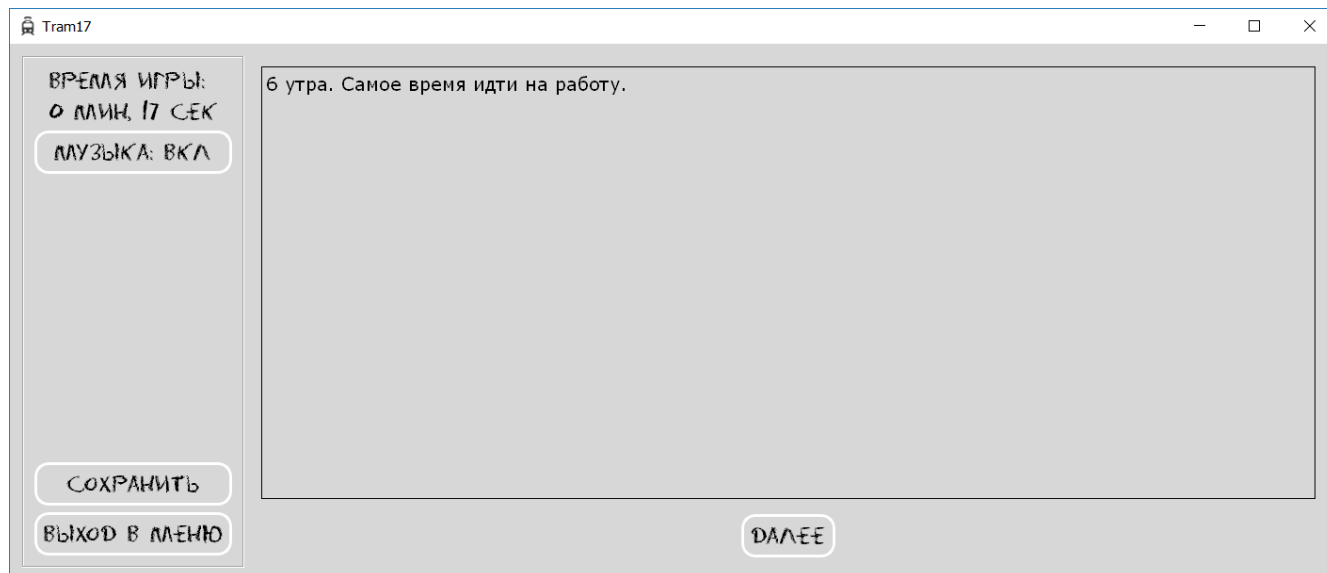
Особенности моего текстового квеста:

- Написан на C++/Qt, без использования различного рода движков.
- Мрачная история и атмосфера.
- Упор на сюжет, а не на количество выборов или решение головоломок.
- Кроме небольших изменений в ходе повествования, также присутствует 7 возможных окончаний истории, правильное из которых найти не так просто.
- Музыкальное сопровождение.
- Достижения.

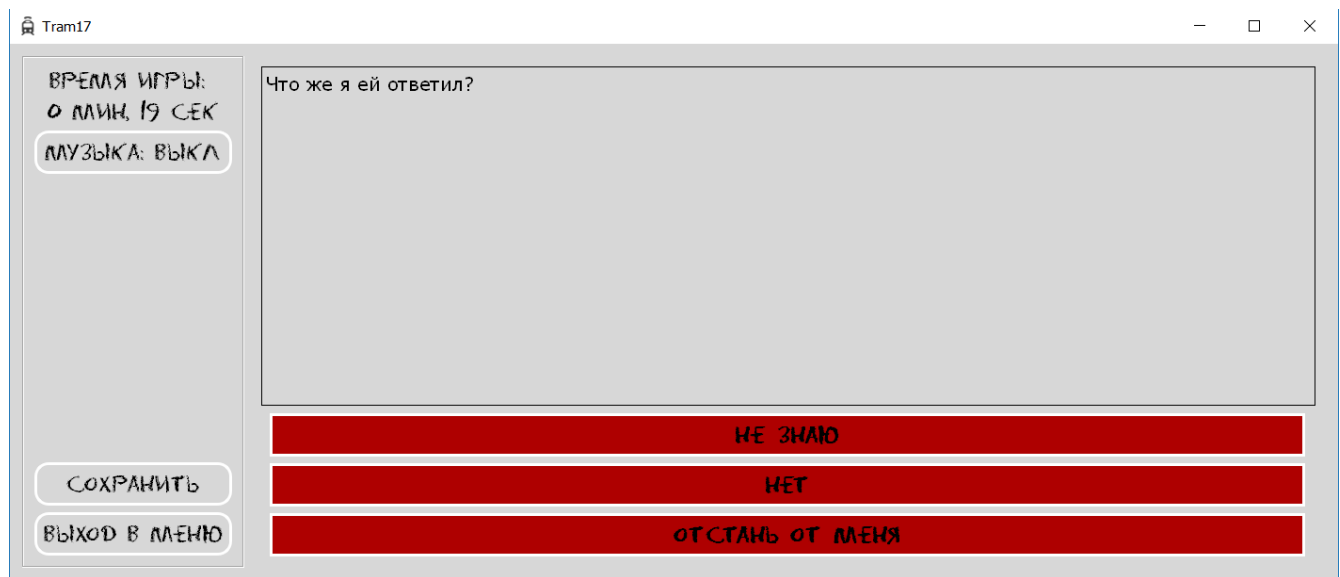
Снимки интерфейса



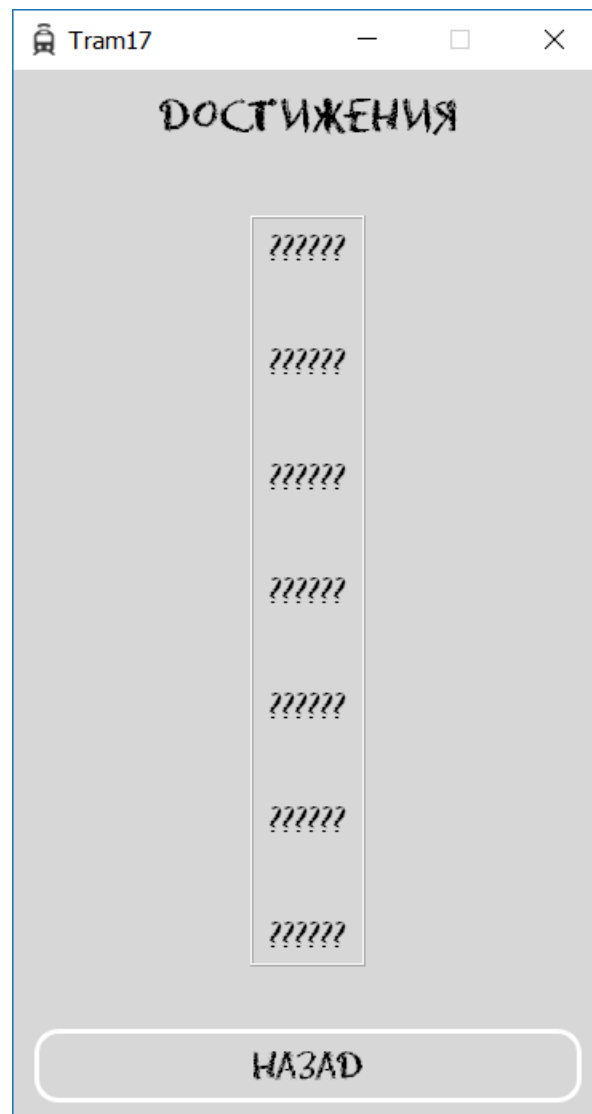
Главное меню игры



Главное окно игры: текстовое окно и элементы управления



Главное окно игры: текстовое окно с кнопками сюжетного выбора



Список достижений игрока

Техническое описание

Хранение данных

Сценарий игры хранится отдельно в обычном текстовом файле «scenario.txt».

Весь текст разбит на фрагменты, отделённые друг от друга пустой строкой.

При старте игры все фрагменты текста последовательно считываются в список(**QStringList**). Затем мы можем манипулировать считанными фрагментами.

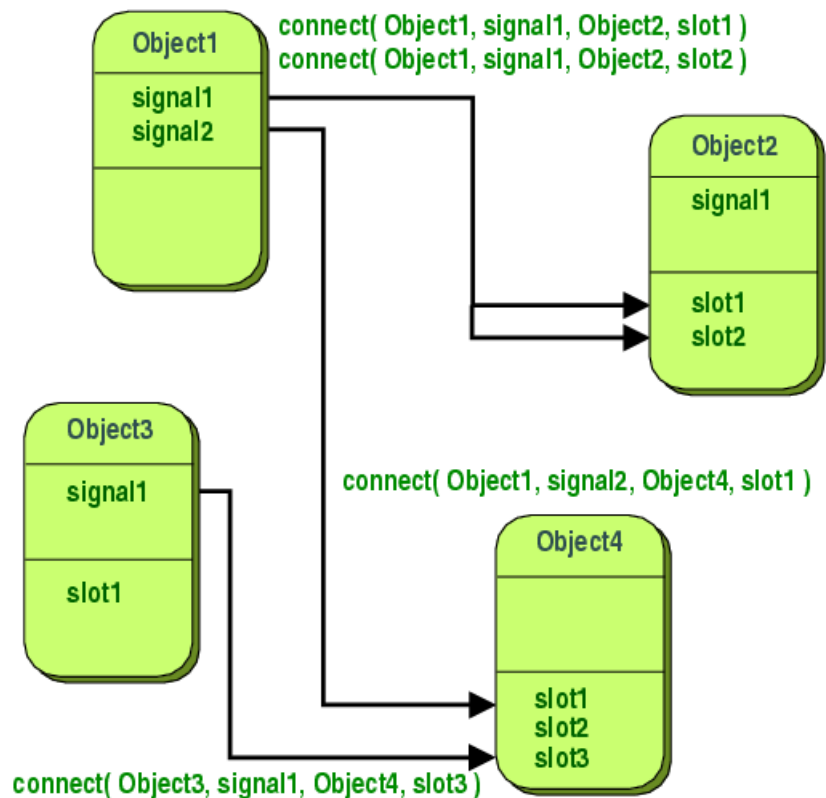
Логика событий

Вся интерактивная часть игры построена на событиях. В библиотеке Qt события реализованы с помощью системы сигналов и слотов.

Сигнал вырабатывается, когда происходит определенное событие.

Слот – это функция, которая вызывается в ответ на определенный сигнал.

С помощью функции **connect** мы соединяем сигнал и слот.



Например:

```
connect(ui->nextButton, &QPushButton::clicked,  
        this, &TextWindow::nextFragment);
```

Теперь при нажатии на кнопку **nextButton** будет вызвана функция **nextFragment**, которая отвечает за смену текста на экране.

Ветвления

В игре присутствуют различные сюжетные выборы, которые можно совершить в определенный момент.

Для их реализации в процессе вывода фрагментов текста на экран программа проверяет, на какой позиции мы находимся. И если эта позиция совпадает с нужной, выводит кнопки выбора на экран и соединяет их с нужными функциями.

Например:

```
...  
if (position == 6)  
    {  
        ui->nextButton->hide();  
  
        choice1->setText("Остаться");  
        choice2->setText("Уйму");  
  
        choice1->show();  
        choice2->show();  
  
        connect(choice1, &QPushButton::clicked,  
            this, &TextWindow::stayEvent);  
  
        connect(choice2, &QPushButton::clicked,  
            this, &TextWindow::leaveEvent);  
    }  
...
```

```
void TextWindow::stayEvent()  
{  
    setText(7);  
    setButtons();  
}
```

```
void TextWindow::leaveEvent()  
{  
    setText(10);  
    setButtons();  
}
```

Сохранение

В игре также реализована система сохранения.

При нажатии на кнопку сохранения или возврате в главное меню текущая позиция в тексте и значения нужных переменных сохраняются в файл.

В дальнейшем сохранение может быть загружено кнопкой «Продолжить».

При старте новой игры сохранение обнуляется.

Достижения

Для каждой из семи концовок реализовано достижение.

Принцип их работы аналогичен сохранению игры.

При наступлении определенного события в игре в файл записывается метка достижения.

При открытии меню достижений метки считываются и сопоставляются с названиями достижений.

Выводы

В игре было реализовано:

- Основа: движок, сценарий, переключение текста.
- Сюжетные выборы.
- 7 возможных окончаний сценария.
- Функция сохранения.
- Достижения.
- Фоновая музыка.

Не было реализовано:

- Множественные сохранения (из-за ненужности данной функции в игре).
- Иллюстрации (по причине отсутствия собственных и нежелания использовать чужие).

Мною были реализованы практически все изначальные задумки.

При создании будущих проектов планируется разработать лучшую систему хранения текста и взаимодействия программы с ним, так как текущая не очень оптимальна; иначе использовать готовые движки.

Большую часть игр подобного жанра делают на специальных движках – системах разработки. У них есть свои преимущества.

Вывод, к которому я пришёл в ходе разработки в том, что для создания текстовых квестов лучше использовать специальные движки, однако, опыт, полученный при разработке квеста с нуля, бесспорно полезен.