

# Постановка задачи проекта ООП "Система диалогов и квестов"

Когтенков Алексей, ПМИ 155 (РС)

## Вводная информация

Во многих компьютерных играх присутствует в том или ином виде система диалогов с персонажами. Обычно она представляет из себя набор ориентированных графов, в которых каждое ребро соответствует вопросу и ответу персонажа на этот вопрос, а вершины соответствуют состоянию диалога. Каждый граф соответствует своему собственному персонажу. В графах поддерживается текущая позиция в диалоге; при переходе по ребру (задании вопроса и получении ответа) позиция меняется, и, соответственно, меняется возможный список вопросов. Ребра в графах могут быть открыты или заблокированы. В случае, если ребро открыто, соответствующий ему вопрос можно задать; если заблокировано – то нельзя. Каждому переходу в диалоге соответствует скрипт, который может быть пустым, блокировать и разблокировать ребра в диалогах, обновлять состояния персонажей и так далее. В общем случае скрипт может делать что угодно.

Однако такая система диалогов не очень реалистична, т.к. игроку показываются все возможные варианты вопросов. Если их много, то это вгоняет игрока в уныние, а если мало, то диалоги просто перестают быть интересными из-за отсутствия выбора. Я придумал улучшение подобной системы и хотел бы реализовать простую версию этого улучшения (полную версию я не успею сделать).

## Постановка задачи

Основой моей диалоговой системы будет система, описанная выше. Однако в нее будут внесены некоторые изменения.

Помимо стандартного графа диалогов добавим так называемые общие вопросы. В нашем упрощенном случае общий вопрос будет определяться двумя ключевыми словами: типом вопроса и объектом вопроса. Например, ключевые слова {где, проектор} определяют вопрос "Где находится проектор?"; {расскажи, ФКН} – "Расскажи о ФКН", и т.д. Несложно понять, что мы по сути получаем матрицу, в каждой ячейке которой находится определяемый ключевыми словами вопрос и ответ персонажа на него. В данной реализации не предполагается разделять объекты вопроса на разные типы (и посему некоторые комбинации ключевых слов могут определять несколько странных вопросы, но это не слишком большая проблема). Поскольку количество общих вопросов сильно растет с ростом числа ключевых слов, то для каждого персонажа имеет смысл хранить **разреженную** матрицу. На вопросы, на которые персонаж не знает ответ, он будет отвечать что-нибудь из серии "не знаю". Здесь стоит отметить, что если персонаж знает ответ на общий вопрос, то этот общий вопрос и ответ являются полноценным ребром в диалоговом графе, и его обработка осуществляется точно так же, как и для обычных вопросов-ответов в графе (и если ребро заблокировано, то даже при наличии ключевых слов вопрос не будет доступен). Ключевые слова тоже могут быть как заблокированы, так и открыты.

У графа диалога с персонажем выделим вершину, назовем ее корневой. С нее будет начинаться любой диалог с персонажем. Для большей простоты и логичности реализации общие вопросы можно задавать, только находясь в корневой вершине.

Как я уже писал выше, каждому ребру будет соответствовать скрипт. В моей реализации скрипты будут уметь блокировать/разблокировать ключевые слова, блокировать/разблокировать переходы в графах, а также выполнять некоторые действия для демонстрационных квестов (скажем, взаимодействовать с инвентарем персонажа). Скрипт будет задаваться в файле на простом псевдоязыке и интерпретироваться во время выполнения.

Все вышеописанное было связано с системой диалогов, теперь перейдем к окружению. Взаимодействие с игровым миром происходит с помощью команд, набираемых в консоли; информация о мире и диалогах тоже показывается в консоли. Предполагается реализовать:

- 1) Обработчик команд для взаимодействия с миром.
- 2) Персонажи, к которым привязаны графы диалогов.

3) Парсеры для считывания информации о диалогах, персонажах и скриптах из файлов (в том числе как методы классов).

4) Модуль ввода.

5) Простейшая система инвентаря для квестов.

**НЕ предполагается:**

1) Работа с базами данных и динамическая подгрузка данных (например, при переходе между локациями). В моей реализации все данные о диалогах будут сразу загружаться, но будет оставлена возможность переписать минимальное количество кода для добавления динамической подгрузки и работы с БД.

2) Какой-либо [удобный] интерфейс разработчика для создания графов диалогов.

3) Хорошая/логичная система хранения данных о диалогах и скриптах. В идеале надо хранить данные в БД (желательно, в сжатом виде), но у меня нет времени, чтобы разбираться с БД в Java или придумывать оптимальную файловую структуру.

4) Сильное наполнение игрового мира контентом. Однако контента будет достаточно для демонстрации основных возможностей диалого-квестовой системы.