

Создание системы отправки голосовых сообщений в реальном времени

Абрамов Владимир Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Лучанинов Дмитрий Васильевич

Приамурский государственный университет имени Шолом-Алейхема

Старший преподаватель кафедры информационных систем, математики и правовой информатики

Аннотация

В работе рассматривается создание голосового чата в реальном времени. Исследование производится в среде разработки Unity3D. В результате был создан прототип программы.

Ключевые слова: Голосовой чат, Среда Unity3D, Обмен голосовыми сообщениями в реальном времени.

Creating a system for sending voice messages in real time

Abramov Vladimir Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Luchaninov Dmitry Vasilevich

Sholom-Aleichem Priamursky State University

Senior lecturer, Department of information systems, mathematics and legal informatics

Abstract

The paper deals with the creation of voice chat in real time. Research performed in the development environment Unity3D. Describes the algorithm for creating Bayesian belief networks.

Keywords: Voice chat, Unity3D Environment, real-time voice messaging.

В наше время очень распространены всякого рода программы-мессенджеры, которые позволяют людям быстро общаться, передавать текстовые и голосовые сообщения. Применение таких программ охватывает не одну сферу жизни. Они применяются для общения в семье, на работе и производстве, на малых и очень больших расстояниях. Мессенджеры работают посредством подключения к глобальной сети интернет. Все они имеют те или иные достоинства и недостатки. Имеют разный интерфейс, но все сделаны с расчётом на то, чтобы пользователю было удобно общаться.

Зачастую мессенджеры имеют стандартный набор функций. Эти функции становятся доступны благодаря подключению к сети Интернет. И естественно, что такую популярность они получили в части и из-за доступности этой сети. Сейчас очень многие знают, что такое Интернет, как им пользоваться, что он может нам дать и чем обеспечить.

Целью данной работы является создание системы отправки голосовых сообщений в реальном времени.

Объектом исследования является система обмена сообщениями.

Предметом исследования является система отправки голосовых сообщений в реальном времени.

Задачи исследования:

1. Создание персонального текстового и голосового чата с одним человеком.

2. Создание групповых текстовых и голосовых чатов (Групп).

3. Совершение аудио звонков, как персональных, так и групповых.

Методы исследования:

- анализ существующих систем;
- выделение общих элементов во всех системах;
- изучения средств с помощью которых создаются данные системы;
- использование выбранных средств для создания собственной системы.

Голосовой чат – это приложение, которое позволяет двум или более пользователям (клиентам) общаться между собой при наличии у всех устройств ввода и вывода звука. Голосовые чаты очень распространены. Они встречаются как основа или дополнение во многих приложениях на разных платформах: мобильных компьютерных и т.д. Например, голосовой чат является основой такого приложения как “Discord” [1], “Skype” [2] или мобильных мессенджеров. Пользователи могут общаться друг с другом напрямую, при этом хостом служат сервера “Skype”.

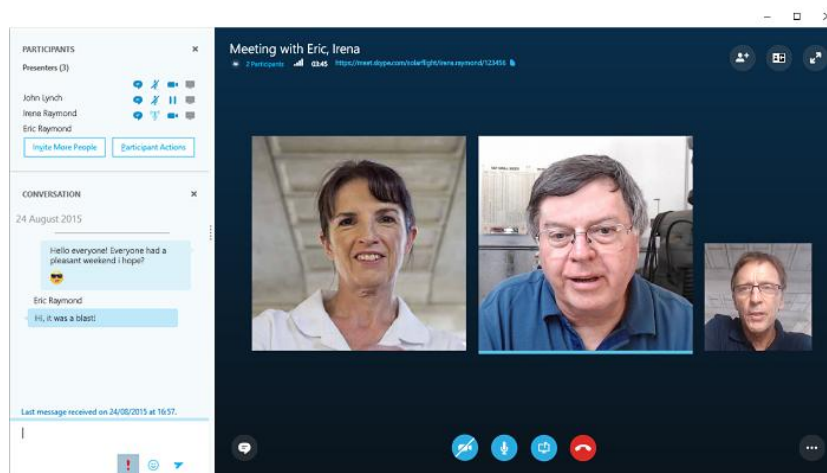


Рис. 1. Пример интерфейса программы «Skype»

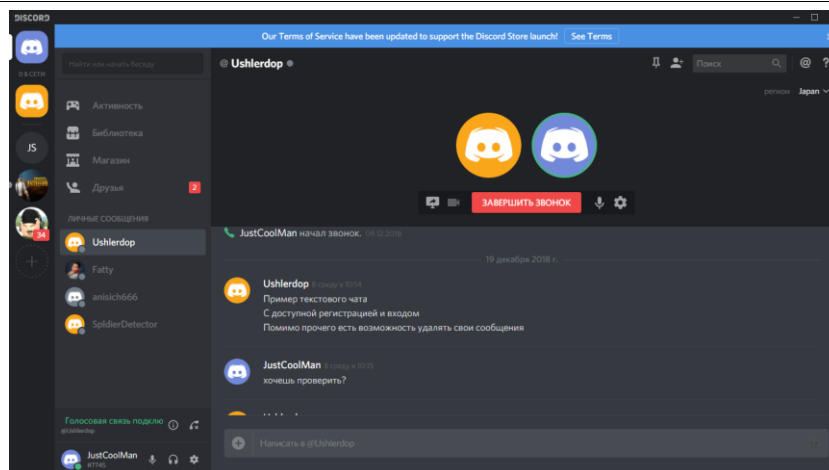


Рис. 2. Пример интерфейса программы «Discord»

Голосовой является дополнением во многих компьютерных, консольных и мобильных развлекательных приложениях, играх, где есть возможность игры по локальной или Интернет-сети. Примером приложений такого рода может служить “Counter-Strike”. При игре по локальной сети один из игроков является сервером (Хостом). Голосовой чат устроен таким образом, что информация, введенная одним из игроков-клиентов, отсылается на сервер (В данном случае к игроку-хосту) и оттуда рассылается по клиентам.

В любом случае схема является понятной и таковой, что в любой системе присутствуют два или более клиента и серверная часть.

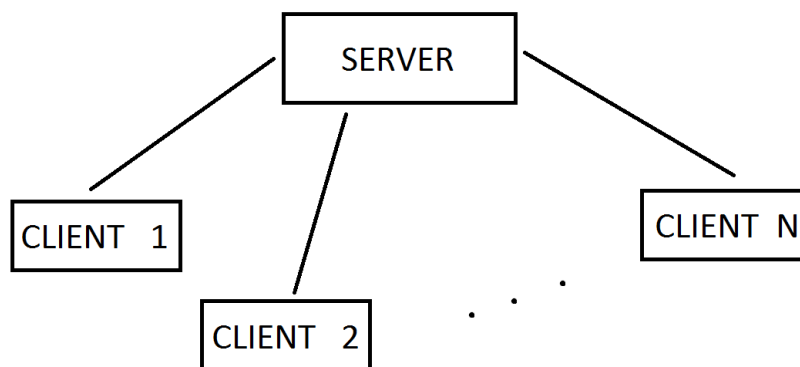


Рис. 3. Сеть при стороннем сервере

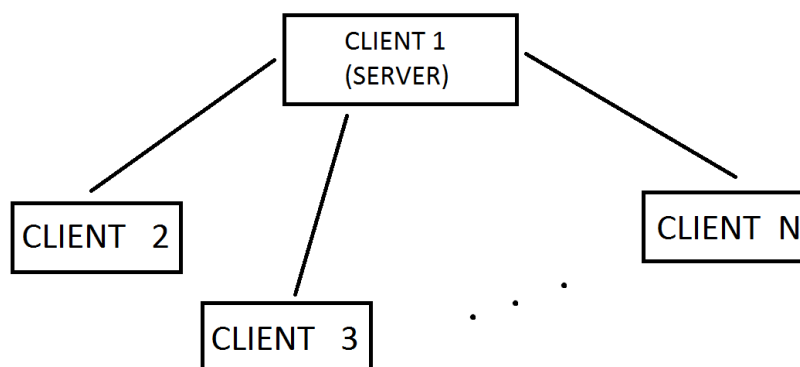


Рис. 4. Сеть при клиенте-хосте

Стоит упомянуть о протоколе передачи сервера. Этот протокол отвечает за перенос сервера с одного клиента на другой без остановки, паузы и перезапуска самого сервера. Это очень удобно в случае разрыва сети у сервера. Однако для осуществления этого протокола клиентам нужно быть соединёнными не только с сервером, но и между самим собой. В случае падения сервера, клиентская часть бывшего сервера может войти на сервер с теми же данными, как простой клиент (рис.5).

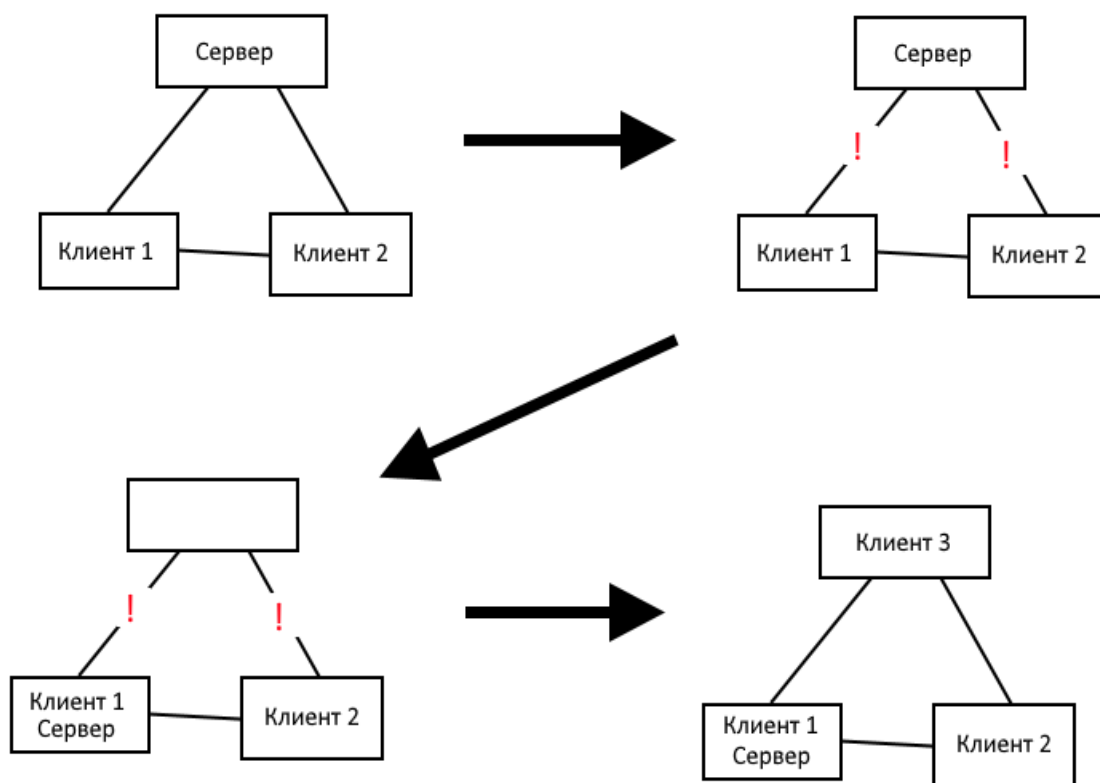


Рис. 5. Протокол передачи сервера

Движок Unity3D [3], на мой взгляд, хорошо подходит для создания и симуляции локальной сети Пользователь(Клиент)-Пользователь(Хост). Движок использует скрипты написанные на “С#”[4]. Встроенные библиотеки UNet, которые были добавлены в Unity, начиная с 5 версии, позволяют создавать сеть напрямую, даже на одном ПК.

В UNet так же есть расширение, которое позволяет выбрать серверную часть приложения. Ее могут выполнять либо сервера “Unity” либо частные клиенты-хосты.

Стоит отметить, что стандартные библиотеки Unity не имеют протокола передачи сервера, а это значит что потеря соединения с сервером приведет к автоматическому отсоединению всех клиентов.

Сама программа, как и любая программа, скомпилированная с помощью Unity3D, состоит из файла «*.exe», Папки с файлами «*_Data», и технического файла UnityPlayer.dll, который адаптирует программу как под 32х-битны, так и под 64х-битные. (Рис.7)

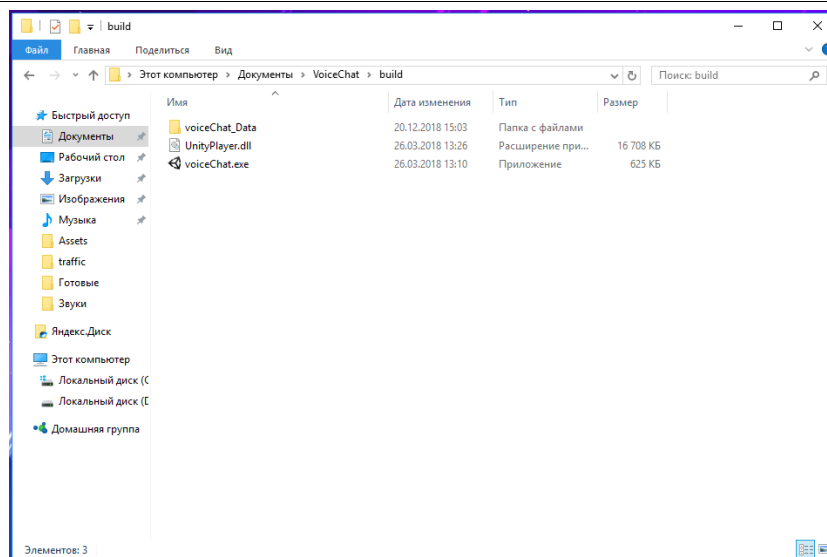


Рис. 7. Комплектация программы

После запуска программы откроется окно – первичное меню (рис.8), в котором есть возможность либо присоединиться к локальному серверу, либо создать новый со своими настройками. (IPv4 и порт). Также есть возможность указать свое имя, которое будет отображаться при разговоре.

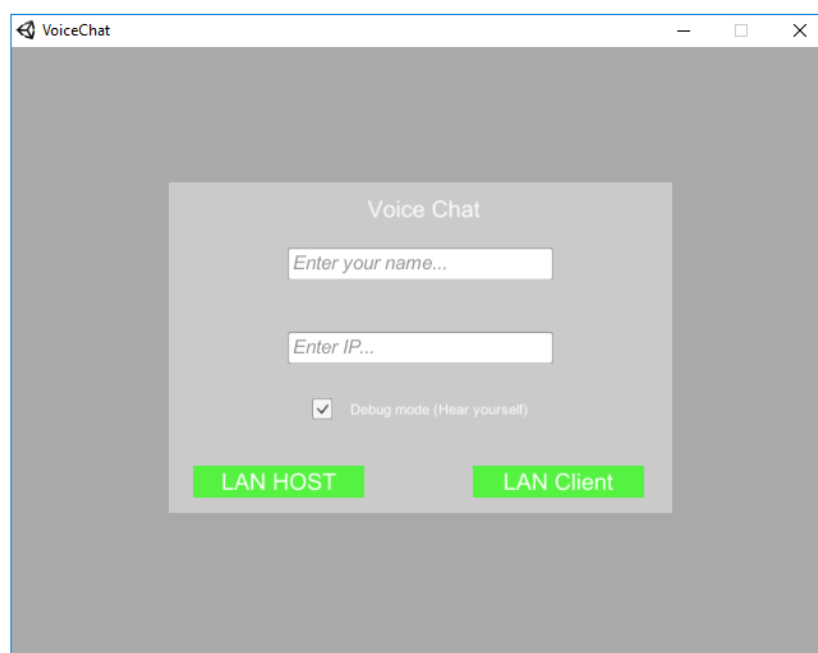


Рис. 8. Первичное меню

Необходимо ввести имя и IP-адрес сервера. Затем его можно будет либо создать как хост, либо подключиться к нему как клиент.

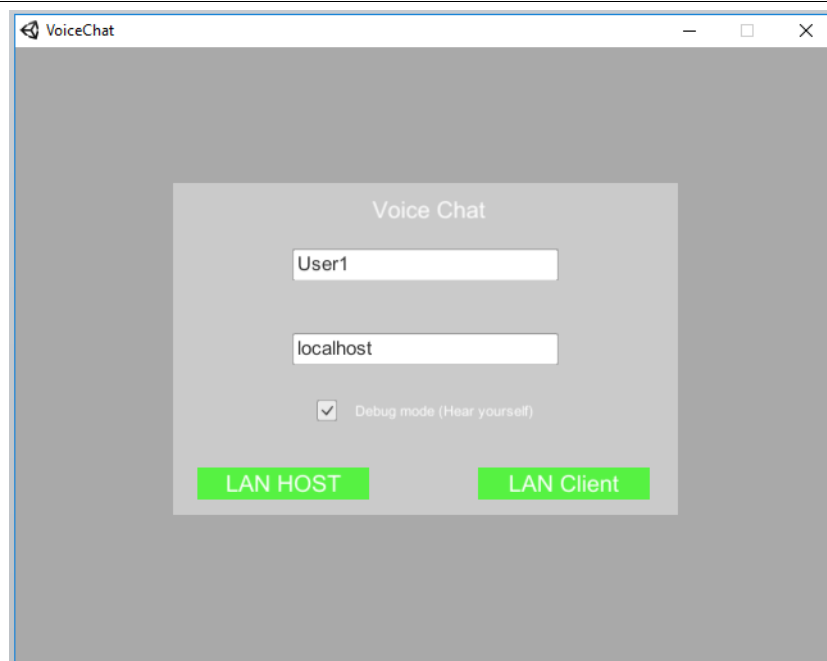


Рис. 9. Пример заполнения первичного меню

При нажатии клавиши «V» (рус. «М») Голосовое сообщение автоматически передается всем клиентам, которые подключены к данному серверу через заданный IP порт.

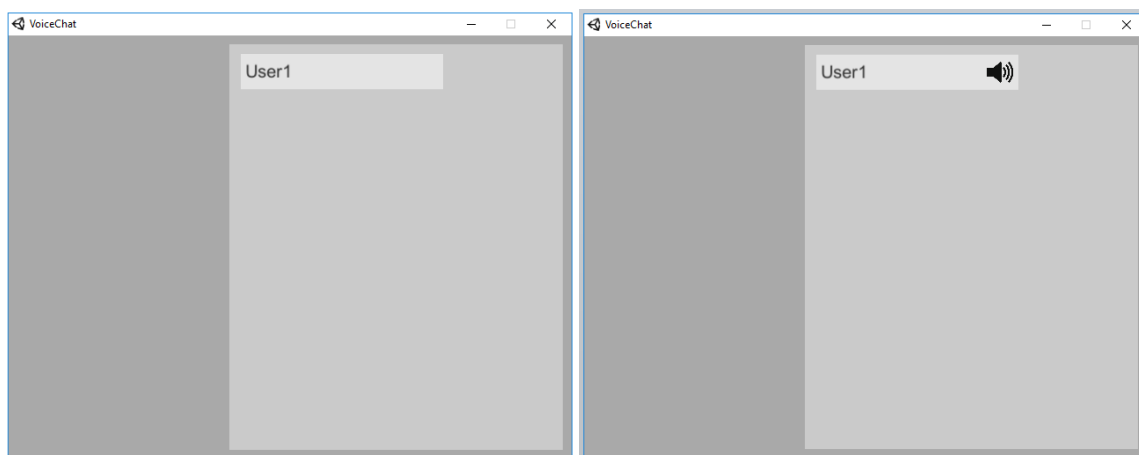


Рис. 10. Окно отображения пользователей

Так же при передаче сообщения появляется маленькая иконка, которая показывает, что этот пользователь говорит в данный момент.

В случае падения сервера все клиенты автоматически отсоединяются и переходят в первичное меню.

Способ, описанный выше, позволяет общаться с людьми в созданной группе. Несмотря на то, что иногда звук передается с незначительными помехами или искажениями, пользователи могут понять всю информацию, прошедшую через группу.

Существуют способы убрать помехи и искажения. Unet и библиотеки Unity всегда дорабатываются и обновляются разработчиками. Правильным

решением для достижения чистоты связи будет использование последних версий этих библиотек.

Текущая версия программы подходит для быстрой и нетребовательной связи между двумя персональными компьютерами.

Библиографический список

1. Discord – официальное приложение. URL: <https://discordapp.com/>
2. Skype – официальное приложение. URL: <https://www.skype.com/ru/>
3. Справочник Unity3D – официальный сайт.
URL: <https://docs.unity3d.com/ScriptReference/>
4. Справочник языка «C#». URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>