# Голосовой чат

В многопользовательском проекте бывает необходимо добавить голосовой чат. Платформа Лямбда-Мю может регулировать настройки, разрешая или запрещая вещание или прослушивание. Однако главной функцией платформы является передача данных между клиентами.

Рассмотрим пример такого проекта.

## Пример реализации

Рассмотрим проект обучения обслуживания серверов: преподаватель и учащиеся подключаются в одну виртуальную комнату - серверную. В режиме 3D или VR. Учащиеся выполняют сценарии обучения или следят за другими пользователями. Всем пользователям доступен общий голосовой чат.

## Репозитарий

Файловая структура проекта представлена ниже.

Необходимые скрипты туториала находятся в репозитарии lm3.examples/godot.voice.

Чтобы запустить платформу скачайте бинарный файл и необходимые библиотеки из репозитария lm3.engine.ce и скопируйте в корень проекта.

## Исходные данные

- Сцена серверной импортированная в Godot Engine
- Запрограммированное управление виртуальным персонажем в режиме 3D и VR
- Запрограммированный модуль связи lm3 и протокол передачи данных

## Цель проекта

Добавление голосового чата с минимальным изменением исходного кода проекта

## Разбор исходных данных

Аудиоданные слишком большие, чтобы не уделять время оптимизации передачи таких данных. В формате CD частота дискретизации 44100 Гц, т.е. 1 секунда аудиоданных весит ~200 кБ.

Чтобы избежать ненужных копирований данных, можно выделить отдельную логическую машину, которая будет транслировать данные от клиента к клиенту, оставив остальные взаимодействия с lm3 на основной машине. На стороне Godot необходимо обработать захват голоса пользователя и прослушивание других пользователей, описать второй TCP клиент и

протокол передачи данных.

## Протокол запросов и ответов

Аудиоданные представляют следующую структуру данных

<code lua> { <ID клиента>, <Данные в байтах>, <Кодек>, <Частота дискретизации>, <Флаг стерео звука>} <code>

## Файловая структура проекта


LM_voice.log
Лог-файл.
voice / start.lua
Скрипт машины.
voice / config.lua
Конфигурационный файл.
Логическая машина (voice)
Основная логическая машина (main)
Сервер логических машин
main / config.lua
Конфигурационный файл.
main / mp_lib.lua
Библиотека объектов.
main / start.lua
Скрипт машины.
LM_main.log
Лог-файл.
lmu3_server.exe
Исполняемый модуль платформы - сервер логических машин.
lm3system.dll
Системный плагин.
lua.dll
Библиотека Lua.
server.log
Лог-файл платформы.
config.lua
Главный конфигурационный файл.

## Скрипты

### voice / start.lua

<code lua> lm3:Log(«Starting LM voice»)

lm3:LoadLibrary( {Alias = «sys», FileName = «lm3system»} )

lm3:CreateLObject( { LibAlias = «sys», Type = «TCPServer», Name = «server» } )
server:SetAttr(«SocCount», lmconf.MaxClientCount) server:SetAttr(«Port», 11024)

server:AddHandler(«OnConnect», function(o, ev)

```
  lm3:Log("OnConnect: "..lm3:DataToString(ev))
```

end)

server:AddHandler(«OnDisconnect», function(o, ev)

```
  lm3:Log("OnDisconnect: "..lm3:DataToString(ev))
```

end)

lm3:AddHandler(«OnStart», function(o, ev)

```
  lm3:Log("OnStart")
```

end)

– трансляция аудиоданных всем, кроме данного пользователя server:AddHandler(«OnIncoming», function(o, ev)

```
  lm3:Log("OnIncoming")
  local array = server:GetAttr("Connected")
  for j = 1, #array do
      if array[j] == true and j ~= ev.Tag then
      --if array[j] == true then
          lm3:Log("send to: "..lm3:DataToString(j))
          server:DoSend({ Tag = j, Data = ev.Data })
      end
  end
```

end) <code>

From:
https://wiki.lambda-mu.com/ - **Lambda-Mu Wiki**

Permanent link:
**https://wiki.lambda-mu.com/lm3/ce/t_godot_voice?rev=1607166040**

Last update: **2020/12/05 14:00**