

Разработка интерфейса пользователя игрового сервера с дополнением database

Разработчики Карманов А. Н.
Косинов И. А.
Рыжков Б. Н.

Студенты учебной группы: К15.1
Научный руководитель: Мухина С.Н.

Введение

Мы рассмотрели актуальные вопросы по созданию игрового сервера, важно учитывать следующие моменты:

- ⦿ для какой игры будет создаваться сервер;
- ⦿ целевая аудитория игрового сервера;
- ⦿ серверное оборудование;
- ⦿ высокоскоростной канал связи, способный выдержать серьезные атаки.

Для разработки интерфейса пользователя игрового сервера использовали следующие программы:

- ⦿ FileZilla
- ⦿ Notepad++

Игровой хостинг

Так как игровой хостинг должен работать стабильно и без перебоев 24/7 и необходима хорошая защита от Dos атак, высокоскоростной канал связи, хороший firewall для блокировки злоумышленников по оборудованию, мониторинг ресурс, оборудование способное выдерживать большую нагрузку и запросы от клиента к серверу.

Лучшим из вариантов был хостинг на MyArena, который является крупнейшим хостингом игровых серверов в России.

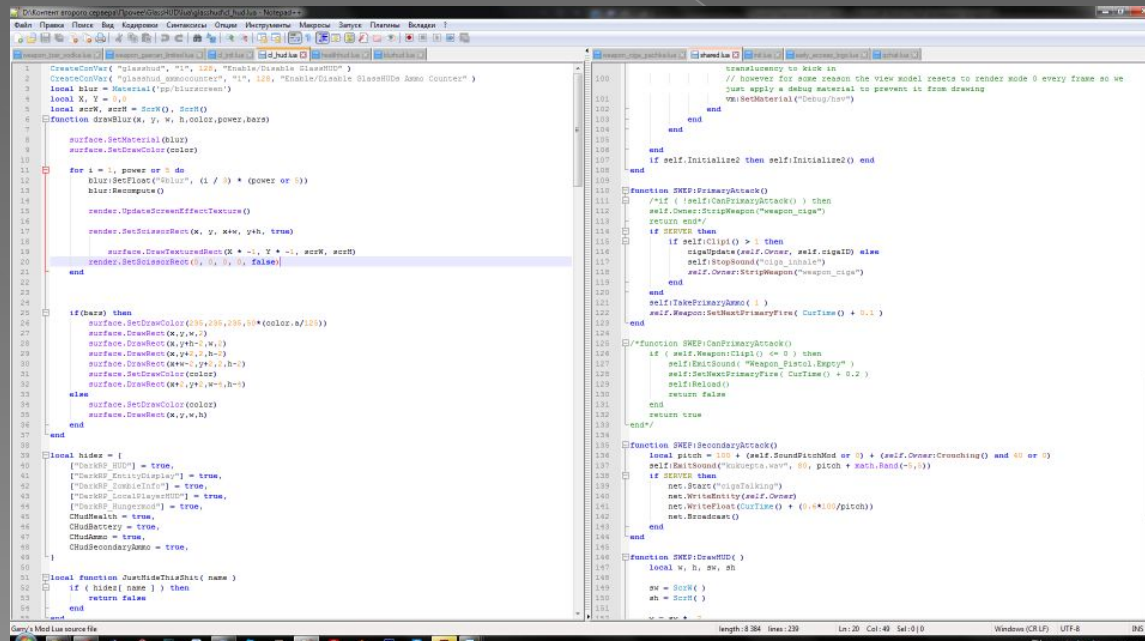
Характеристики сервера

- Процессор: 2 x Core i7 7700K;
- Оперативная память: 4Gb RAM;
- Дисковая квота: 60Gb SSD;
- OS: Linux Ubuntu.

Язык программирования

- На игровом сервере используется скриптовый язык программирования Lua.
- Lua не содержит понятия класса и объекта в явном виде легко реализуются с использованием метатаблиц.
- Lua часто используется для написания скриптов в играх ссылаясь на их библиотеку.

Пример скрипта на языке Lua



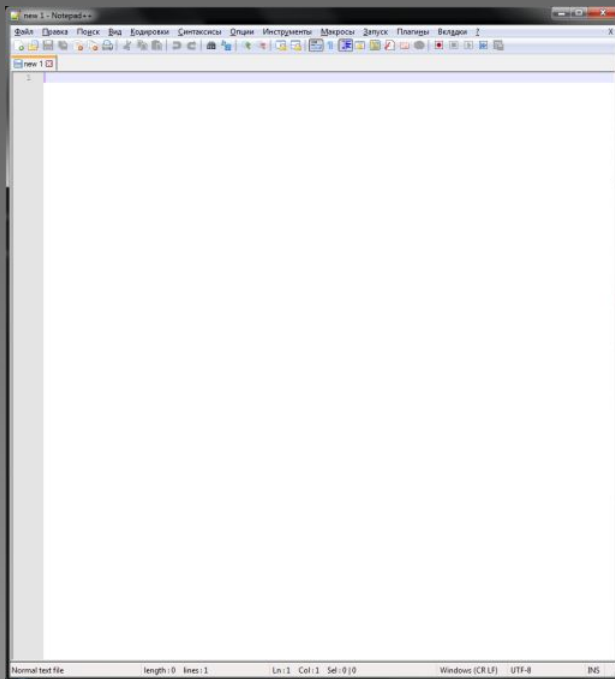
```
1 CreateConsole("classhud", "1", 128, "Enable/Disable classHUD")
2 CreateConsole("classhud_ambocounter", "1", 128, "Enable/Disable classHUD Ambocounter")
3 local blur = {}
4 local X, Y = 0, 0
5 local work, workH = 0, 0
6 function drawBlur(x, y, w, h, color, power, bars)
7
8     surface.SetMaterial(blur)
9     surface.SetDrawColor(color)
10
11     for i = 1, power or 3 do
12         blur:SetFloat("Blur", (i / 3) * (power or 3))
13         blur:Recalculate()
14     end
15     render.UpdateScreenEffectTexture()
16
17     render.SetScissorRect(x, y, work, workH, true)
18
19     surface.DrawTexturedRect(x - 1, y + 1, work, workH)
20     render.SetScissorRect(0, 0, 0, 0, false)
21 end
22
23
24
25 if (bars) then
26     surface.SetDrawColor(255, 255, 255, 50 * (color.r / 255))
27     surface.DrawRect(x, y, w, h)
28     surface.DrawRect(x, y, w, h)
29     surface.DrawRect(x, y, w, h)
30     surface.DrawRect(x, y, w, h)
31     surface.SetDrawColor(color)
32     surface.DrawRect(x, y, w, h)
33 else
34     surface.SetDrawColor(color)
35     surface.DrawRect(x, y, w, h)
36 end
37
38
39 local hider = {
40     ["classHUD_HUD"] = true,
41     ["classHUD_Ambocounter"] = true,
42     ["classHUD_SomeInfo"] = true,
43     ["classHUD_LockDisplayHUD"] = true,
44     ["classHUD_HunterHUD"] = true,
45     CHudSearch = true,
46     CHudAccuracy = true,
47     CHudAmmo = true,
48     CHudSecondaryAmmo = true,
49 }
50
51 local function JustHideThisThis(name)
52     if (hider[name]) then
53         return false
54     end
55 end
```

```
100     transparency to kick in
101     // however for some reason the view model seems to render mode 0 every frame so we
102     just apply a debug material to prevent it from drawing
103     mat:SetMaterial("Debug/NoV")
104 end
105
106 end
107
108 if self:Initialized() then self:Initialize() end
109
110
111 function SWEP:PrimaryAttack()
112     if ( !self:CanPrimaryAttack() ) then
113         self:DamageStripWeapon("weapon_rifle")
114     end
115 end
116
117 if SERVER then
118     if self:Clip() > 0 then
119         self:Update(self:Owner, self:Clip() * 0.1)
120         self:Sound("vfx_bullet")
121         self:Owner:StripWeapon("weapon_rifle")
122     end
123 end
124
125 self:TypeSecondaryAttack()
126 self:Weapon:SetNextPrimaryFire(CurTime() + 0.1)
127 end
128
129
130 function SWEP:SecondaryAttack()
131     if ( self:Weapon:Clip() <= 0 ) then
132         self:Sound("vfx_bullet")
133         self:SetNextPrimaryFire(CurTime() + 0.2)
134     else
135         return false
136     end
137     return true
138 end
139
140
141 function SWEP:SecondaryAttack()
142     local pitch = 100 + (self:SoundPitchMod() * (self:Owner:Crouching() and 60 or 0))
143     self:Sound("vfx_bullet", 80, pitch * math.Rand(-1, 1))
144     if SERVER then
145         net.Start("vfx_bullet")
146         net.WriteEntity(self:Owner)
147         net.WriteFloat(CurTime() + (0.1 * math.Rand(-1, 1)))
148         net.Broadcast()
149     end
150 end
151
152 function SWEP:DrawHUD()
153     local w, h, dx, dy
154     dx = 0
155     dy = 0
156 end
```

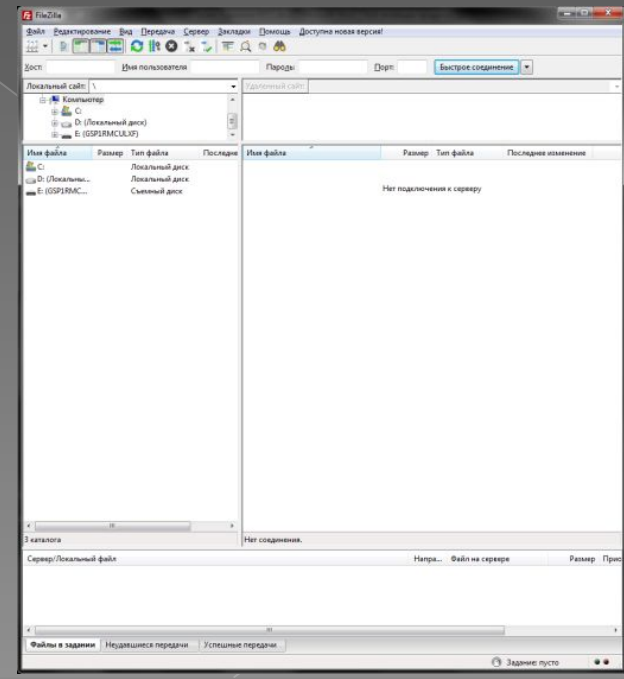
Используемые программы

- FileZilla - это свободный многоязычный FTP-клиент с открытым исходным кодом для Microsoft Windows, Mac OS X и Linux.
- Notepad++ - свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки.

Notepad++



FileZilla



Проделанная работа

В процессе работы были написаны скрипты для игрового сервера, которые:

- Ссылаясь на запрос клиента к серверу, выводят на экран игровую статистику – время, проведенное в игре, назначение статуса клиенту, которое позволяет отслеживать выполнения правил игры клиентами, предоставления дополнительных функций игры;
- Информацию о клиенте, которая позволяет отслеживать этапы игры;
- Запись всей информации о клиенте в базу данных сервера для сохранения информации, полученной в течение игры.

Вывод

В процессе реализации интерфейса пользователя были сделаны следующие действия:

- Был написан скрипт для улучшения взаимодействия клиента с сервером.
- Ознакомление с языком программирования Lua и изучение библиотеки функций для написания скриптов.
- Получение навыков работы с программами FileZilla и Notepad++

