

[DRAFT]

Customisation of Zombie Riot on i3D server #04 [213.163.69.138:27015]

Dan “Chimera” Cassey

August 6, 2011

1 Introduction

This document outlines the Specification and Design for the modification to the i3D Zombie Riot server as suggested by Rob.

2 Background Information: Night of the Dead server

Rob’s suggestions stem from a current server, located in the United States of America, called “Night of the Dead”. The server runs Zombie Hell mod, the precursor to Zombie Riot, written for EventScripts. The server has been heavily customised, including a human class system, a currency system and player character customisation. Player customisation includes changing the character model, adding motion trails behind the player and adding player glow effects.

3 Specification

1. This project will be in the form of a plug-in for the SourceMod[1] scripting platform, written in SourcePawn and using SQLite for local persistence.
2. This plug-in will be running on a Source Dedicated Server on Linux running Counter-Strike: Source
3. This plug-in will run in parallel with the Zombie Riot mod by Greyscale[2].
4. This plugin shall implement a human class system with the following classes:

Assault Allows the player run faster.

Sniper Allows the player to cause more damage when using sniper rifles.

Medic Allows the player to heal team mates.

Support Gives the player 200% more bullets per magazine for the M249 light machine gun.

These are the current classes on the Night of the Dead server. We may add further classes in the future as deemed necessary.

5. This plugin shall allow players to customise thier in-game character. This is to be implemented with a !shop menu where players can buy items for thier character and a credit system where players earn credits *a)* for in-game events such as joining the server, remaining on the server for a prolonged period of time and getting kills; or *b)* by selling previously purchased character modifications.

4 Design

4.1 Source Code Structure

TODO

4.2 Database

The database used for persistence will consist of a single table for storing the attributes of individual players. This table will use external integer IDs which will refer to entities defined on configuration key-value files. The diagram shown right details the sqlite table design specification. The primary key, Client ID, should be auto-generated by sqlite but shall not be used by the system. A player will be identified in the table using their unique Steam ID (in the form STEAM_0:X:XXXXXXXXX where the Xs are integers between 0 and 9). Total credits is the amount of currency, this player can spend on customisations. Class ID, Trail ID, Glow ID, and Player-model ID are integers that correspond to classes, trails, glows and player models respectively defined in the KV configs.

Players	
•client_id	int
•steam_id	text
•class_id	int
•total_credits	int
◦trail_id	int
◦playermodel_id	int

Figure 1: Design of the single table of the database showing field attributes

4.3 Key-Value (KV) Configuration Files

The KV config files will allow administrator users of the system to define classes and player customisations for their own needs. This system will use 4 KV files: classes, trails, glows and models. When parsing these files, certain conditions need to be met, otherwise the system will log an error and disable the affected module of the system. Example conditions: all identifiers must be unique and non-null, all modifiers must be greater than zero and non-null.

4.3.1 classes.kv

The Classes KV will define the identifiers and modifiers of the human classes. Each entry will contain the following fields:

class_id The unique identifier of this class.

class_name The name of this class, used for menus and messages.

is_healer_class Boolean value indicating if this is a medic class.

medic_medpacks Number of medpacks available to give in a round. Only applicable when **is_healer_class** is set to true.

modifier_health The modifier for the starting health of this class.

modifier_speed The modifier for the player speed of this class.

modifier_damage_rifle The modifier for the damage done by assault rifles

modifier_damage_sniper The modifier for the damage done by sniper rifles

modifier_damage_lmg The modifier for the damage done by the light machine gun

modifier_damage_smg The modifier for the damage done by sub machine guns

modifier_damage_shotgun The modifier for the damage done by shotguns

modifier_bullets_rifle The modifier for the number of bullets given to a player when purchasing an assault rifle.

modifier_bullets_sniper The modifier for the number of bullets given to a player when purchasing a sniper rifle.

modifier_bullets_lmg The modifier for the number of bullets given to a player when purchasing the lmg.

modifier_bullets_smg The modifier for the number of bullets given to a player when purchasing a sub machine gun.

modifier_shells_shotgun The modifier for the number of shells given to a player when purchasing a shotgun.

modifier_magazine_rifle The modifier for the size of assault rifle magazines.

modifier_magazine_sniper The modifier for the size of sniper rifle magazines.

modifier_magazine_lmg The modifier for the size of light machine gun magazines.

modifier_magazine_smg The modifier for the size of sub machine gun magazines.

modifier_magazine_shotgun The modifier for the size of shotgun magazines.

4.3.2 trails.kv

The Trails KV will define the identifiers and effect files for the trails. TODO

4.3.3 glows.kv

The Glows KV will define the identifiers and effect files for the glows. TODO

4.3.4 models.kv

The Modes KV will define the identifiers and model file paths for the player models. TODO

References

- [1] SourceMod: Half-Life 2 Scripting, <http://www.sourcemod.net/> [Online], accessed August 6, 2011.
- [2] Zombie Riot - Allied Modders, <http://forums.alliedmods.net/showthread.php?p=647040> [Online], accessed August 6, 2011.



Valve, the Valve logo, Counter-Strike, the Counter-Strike logo, Source and the Source logo are trademarks and/or registered trademarks of Valve Corporation.