That's a great idea for documentation! A **Man (Manual) page format** is perfect for a CLI-based game, offering concise and structured help.
Here is the database of core **NexusScript** commands and syntax in a simplified Man-page format, broken down by function and tiered by player progression.

NexusScript Command Database (Man Format)

I. Core System Commands (Level 1)

MAN PAGE: ls

NAME **ls** - list contents of virtual directories
SYNOPSIS ls [path]
DESCRIPTION Displays files and sub-directories within the **Virtual Computer (VC)** file system at the specified path. If no path is provided, lists the current directory.
OPTIONS -l (long) Shows file permissions and size (e.g., rwx, 32kb).
EXAMPLES ls /v system ls -l /v modules

MAN PAGE: cat

NAME **cat** - print contents of a file
SYNOPSIS cat [filepath]
DESCRIPTION Reads and outputs the contents of the specified virtual file to the **NexusShell** console.
EXAMPLES cat /v logs auth.log cat /v data passwords.txt

MAN PAGE: set

NAME **set** - declare and assign a variable or object
SYNOPSIS set $[variable] = [value]
DESCRIPTION Creates a variable and assigns it a value (String, Integer, Boolean) or instantiates a new **NexusScript** object. Variables must be prefixed with $.
EXAMPLES set $user = "admin" set $target = new IP 10 0 0 5 set $online = TRUE

MAN PAGE: print

NAME **print** - display text or variable contents
SYNOPSIS print [message] [$[variable]]
DESCRIPTION Outputs the specified text string or the value of a variable to the console.
EXAMPLES print "Scanning target network..." print "Password found: " $pass

II. Basic Networking and Scripting (Level 1–5)

MAN PAGE: ping

NAME **ping** - test network connectivity
SYNOPSIS ping [IP_Object or address]

DESCRIPTION Sends a network request to the target to determine if it is alive and calculates virtual **latency**.
EXAMPLES ping 10 0 0 1 ping $router_ip

MAN PAGE: scan

NAME **scan** - perform network discovery and port enumeration
SYNOPSIS scan [IP_Object or address]
DESCRIPTION Identifies open ports and services running on the target. Returns a collection of **Port_Object** results. Speed is limited by **NIC Hardware**.
OPTIONS --service Attempts to identify the specific software version of the running service. --subnet Scans the entire subnet range (e.g., scan 192 168 1 0 --subnet 24).
EXAMPLES scan $target_ip --service set $ports = scan 10 0 0 5

MAN PAGE: run

NAME **run** - execute a saved NexusScript module
SYNOPSIS run [module_name] ([arguments])
DESCRIPTION Executes a player-created function (module). Arguments are passed to the module's parameters.
OPTIONS --debug Performs a dry run, checking for basic syntax errors before execution.
EXAMPLES run my first script run brute force v1 ($server_ip $wordlist_file)

III. Object Method Reference (Common)

MAN PAGE: [Object].connect

NAME **[Object].connect** - attempt to establish a session with a target service
SYNOPSIS $[Service_Object].connect([username] [password])
DESCRIPTION Attempts to log in to the specified service. Used on **Service_Object** instances found via scan. Returns TRUE on success, FALSE on failure.
EXAMPLES $ftp_21.connect("anonymous" "guest@") if $target.connect($user $pass_var) { print "Success!" }

MAN PAGE: [Object].login

NAME **[Object].login** - (alias for connect) attempts login to a discovered service
SYNOPSIS $[Service_Object].login([username] [password])
DESCRIPTION See **[Object].connect**.
EXAMPLES set $status = $web_server.login($u $p)

IV. Advanced Hacking (Level 10+)

MAN PAGE: hashcrack

NAME **hashcrack** - crack a password hash using a wordlist

SYNOPSIS hashcrack [hash_string] [Wordlist_Object]
DESCRIPTION Attempts to find the plaintext password for a captured hash. Time taken depends on the **CPU Cores** and the complexity of the **Wordlist_Object** (ruleset).
OPTIONS -a (algorithm) Specify the simulated hashing algorithm (e.g., md5, sha256). Default is automatic detection.
EXAMPLES hashcrack "ab12c3" $common_list set $p = hashcrack $captured_hash -a sha256

MAN PAGE: pivot

NAME **pivot** - use a compromised asset as a proxy jump-point
SYNOPSIS pivot [Compromised_IP] [command] [target]
DESCRIPTION Routes a subsequent network command through a successfully compromised server (where root access has been gained) to access an otherwise unreachable internal network segment.
EXAMPLES pivot $server_1 scan 10 10 10 5 pivot $internal_router ping 172 16 0 2

MAN PAGE: [Exploit_Object].deploy

NAME **[Exploit_Object].deploy** - execute a vulnerability exploit
SYNOPSIS $[Exploit_Object].deploy([target IP] [target OS] [payload module])
DESCRIPTION Deploys the code contained within the Exploit_Object against the target. Requires the correct target system/service version. Success results in gaining **root** access.
EXAMPLES $cve_77.deploy($target_ip "win_server_v2" $shell_payload)
$zero_day_v1.deploy($unknown_target "auto" $data_exfil)