VIP APARTAMENT

USER MANUAL

• ABOUT THE DEMO SCENE •

The demo scene «VIP Apartments» demonstrates the capabilities of my game engine Native Engine 0.2.x, on which I plan to make good, high-quality games, and is a small quest. I will implement everything myself — from the initial idea to promotion.

• STORY •

The morning after a hectic party, you wake up in your penthouse all alone. Your head is splitting with a hangover and you don't remember anything from yesterday. However, you are awakened by an SMS from a hidden number, which says that you must find and return the thing that was hidden due to drunkenness. But remembering the events that happened yesterday is still a challenge! The only clue is a note that someone has carefully left on the coffee table next door.

From this point on, your quest begins. You can open and close any door, turn on and off any device, and take whatever you see.

• SYSTEM REQUIREMENTS •	
64-bit OS	Linux (X11) or Windows 7+
Processor	Multi-core Intel or AMD
RAM	8 Gb
Video card	With 4 Gb VRAM and OpenGL 4.3 support
Sound card	Integrated or discrete
Disk space	500 Mb

The Native Engine runtime uses only cross-platform libraries and does not require any additional software such as Visual C++ Runtime, DirectX and PhysX. Also, in the engine runtime, only cross-vendor technologies are used, and therefore it should work equally on computer components from different vendors.

• CONTROL •	
Move	W, A, S, D
Camera control	Mouse
Sprint	Shift
Jump	Space
Crouch	Alt
Use	Left Mouse Button
Save screenshot	F12
Windowed / fullscreen mode	Alt + Enter
Quit	Esc or Alt + F4

Interaction with most game objects is done either by clicking the Left Mouse Button, or by holding down the Left Mouse Button and moving the mouse. The application protocol and screenshots are saved in the «Documents / VIP Apartament» and «Images / VIP Apartament» folders, respectively.

You can follow further developments in my blog: https://vk.com/native.engine (so far only in Russian). There you can also support future projects with a small donation.

• SOFTWARE USED •

OS Ubuntu, Google Chrome, OnlyOffice, Double Commander, Hyper Terminal, Code::Blocks and Glade, GCC and Mingw-w64, CMake, Atom (scripts and shaders), Blender, GIMP, Inkscape, Scribus, Audacity

• LOGOS •

• MATERIAL USED •

Sonniss.com • GDC • Game Audio Bundle

• SOFTWARE DEVELOPMENT KITS USED •

Linux Kernel System Calls, X11, Windows API, OpenGL and GLSL,
OpenGL Extension Wrangler Library, OpenGL Mathematics,
OpenAL Soft and Ogg Vorbis, FFmpeg and dav1d (decoder AV1),
Bullet Physics Library, Programming Language Lua,
Subpixel Morphological Antialiasing, Ambient Occlusion Volumes,
zlib, libpng, TinyEXR, TinyXML-2

• LOGOS •

• ALSO USED IN THE TOOLKIT •

GTK, Open Asset Importer Library, FreeImage, FreeType