

KNOX

# N64 Development

GAME

Knox Game Design

March 2026

DESIGN

# Architecture

- VR4300 CPU
  - Modified NEC VR4300
  - FPU (floating point unit) - COP1
- Reality Coprocessor
  - Reality Signal Processor
  - Reality Display Processor
- RDRAM – Rambus DRAM 4 MiB
- PIF-NUS
  - based on Sharp SM5 microcontroller
- Audio DAC (digital to analog converter), Video DAC

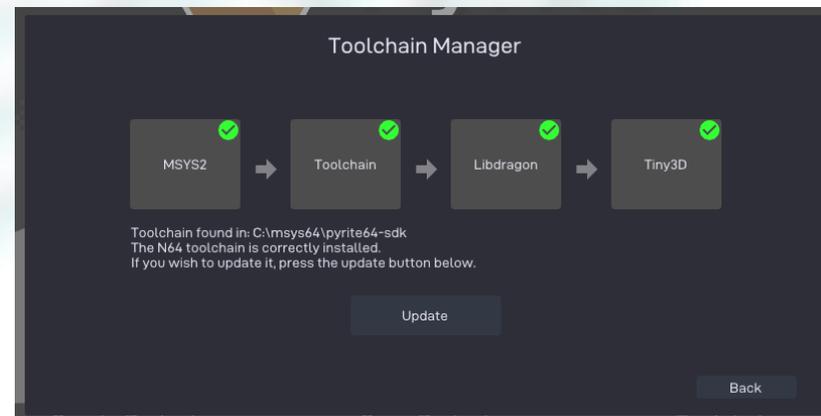
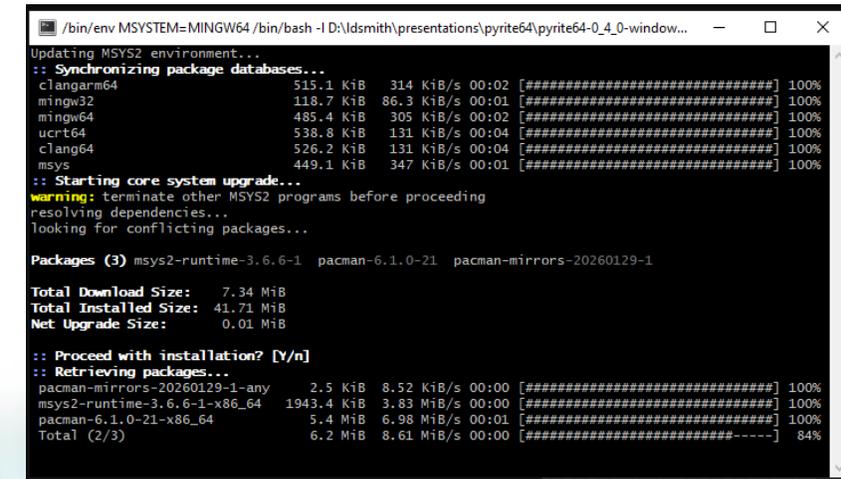
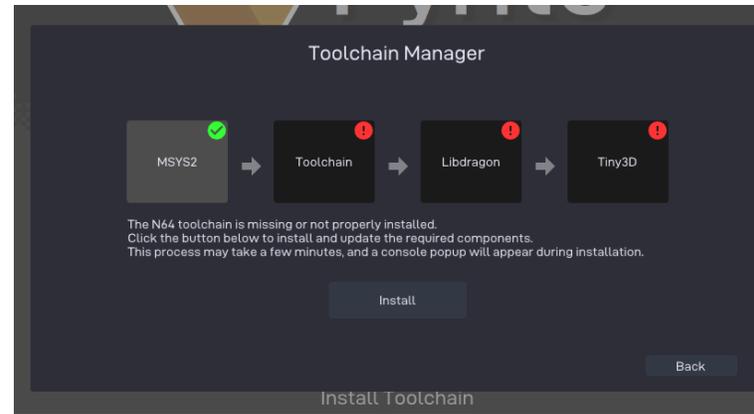
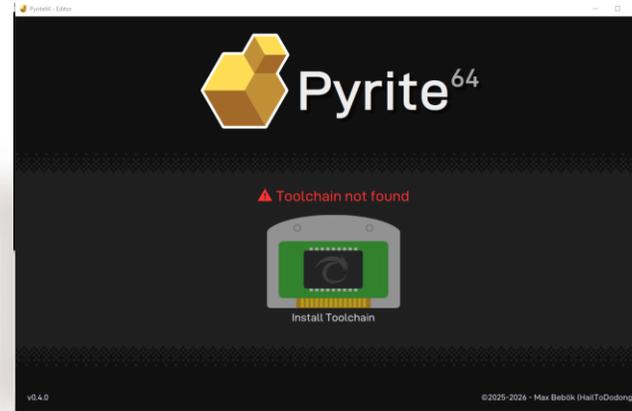


# Development Environment

- Game development environment for N64
  - Like Unity or Godot
  - Only authentic emulators
  - Can run on real N64 hardware
- Download from SourceForge or GitHub
- Windows and Linux builds / Source Code

# Installing (Windows)

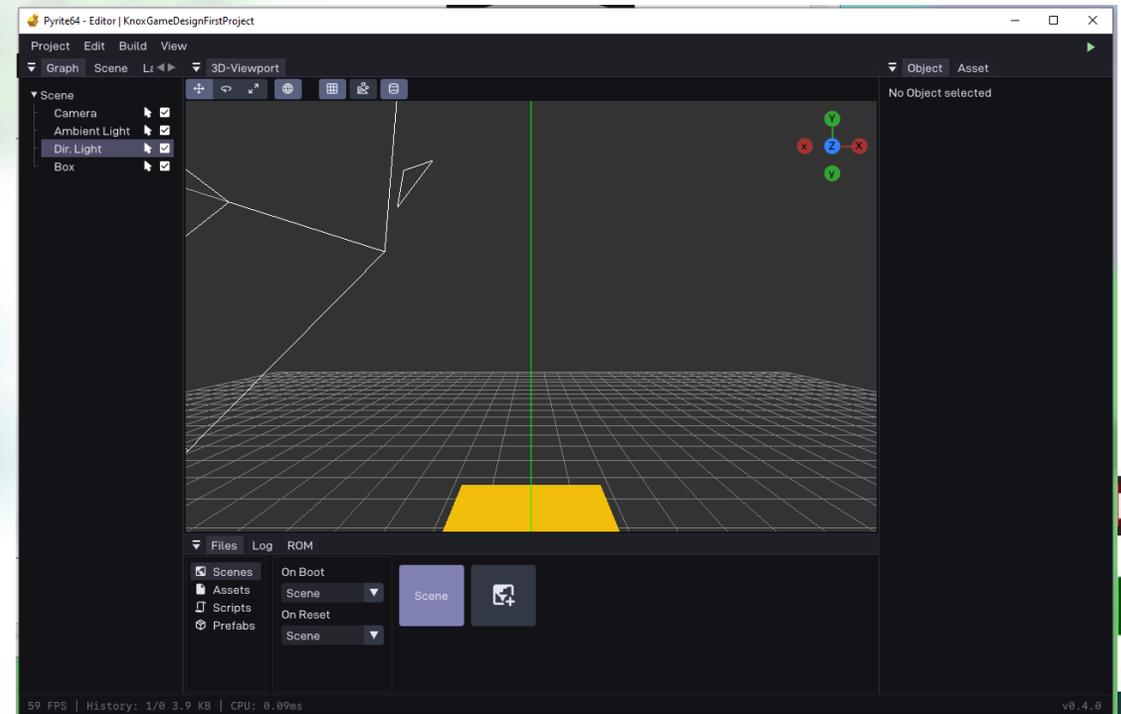
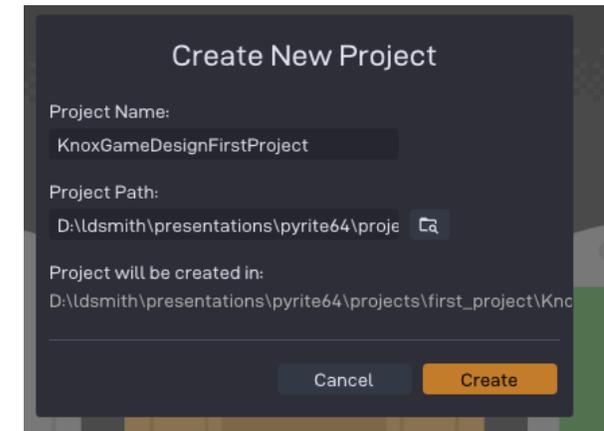
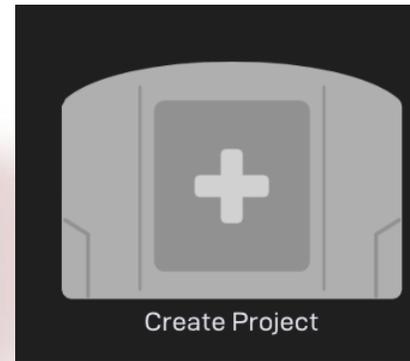
- Download and extract the ZIP
- Run pyrite64.exe
- Install the toolchain
  - MSYS2
    - must use c:\msys64 as install directory
    - Having existing MSYS2 installation at d:\msys64 doesn't work
  - Press install to install the toolchain



**KNOX**  
**GAME**  
**DESIGN**

# Create new project

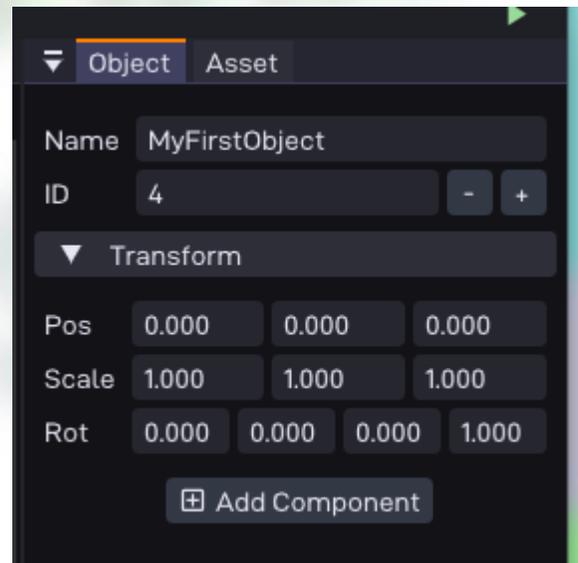
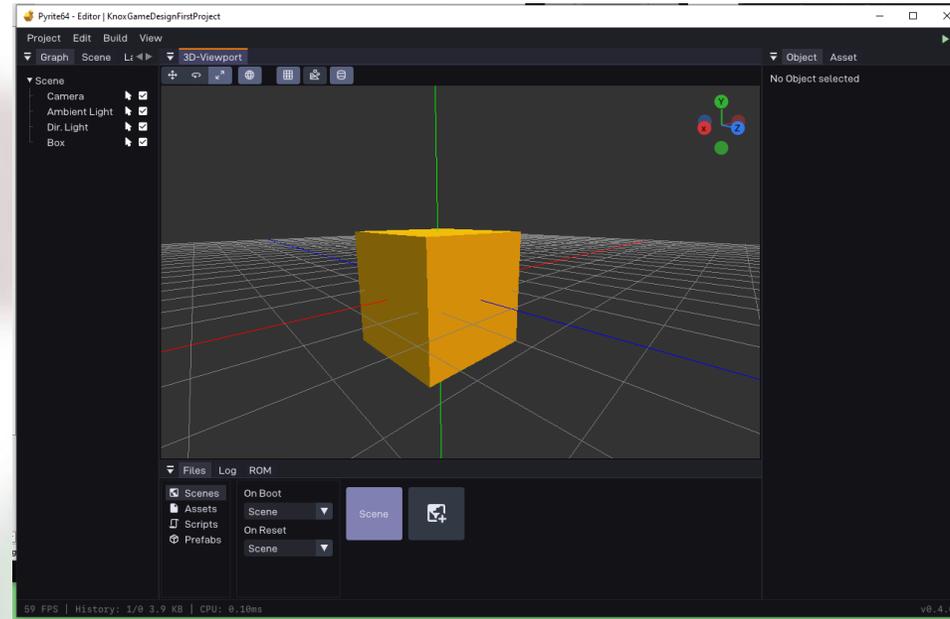
- Select Create New Project
- Select a project folder



X  
E  
DESIGN

# Editor

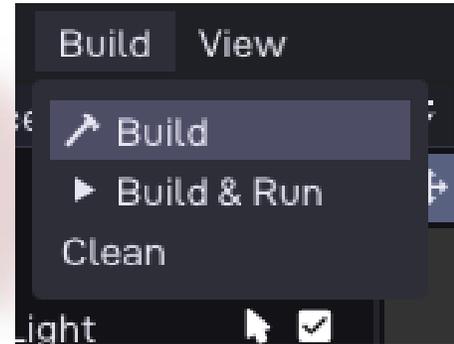
- Hold right mouse / WASD to move view
- Right click Scene > Add Object
- Add Component
- Model (Static)
- Model box.glb



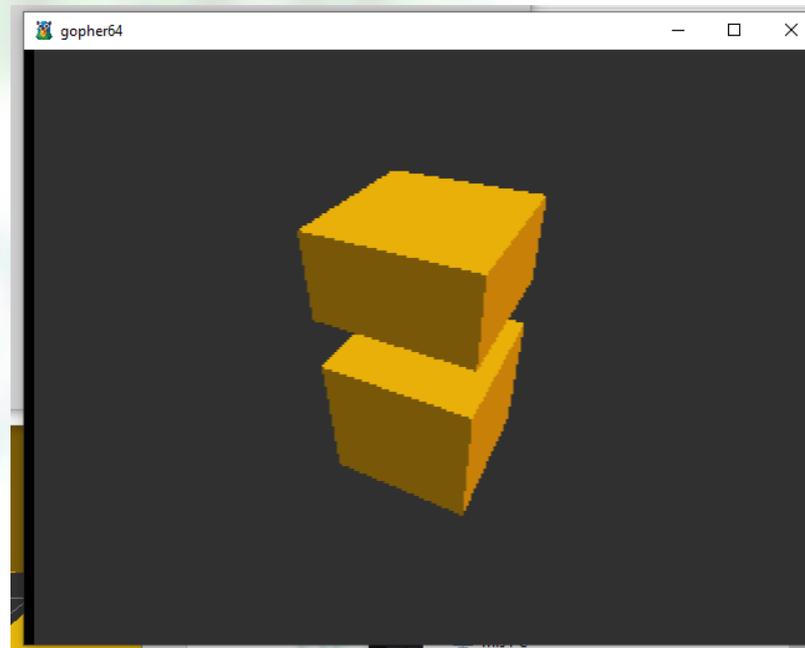
**KNOX**  
**GAME**  
**DESIGN**

# Creating a build

- Build > Build and Run
- Ares emulator must be downloaded separately in Windows
  - Can set ares executable in Project > Settings > Emulator so that it starts automatically on Build & Run
- Creates a .Z64 file in the project directory
- Can run with gopher64 emulator



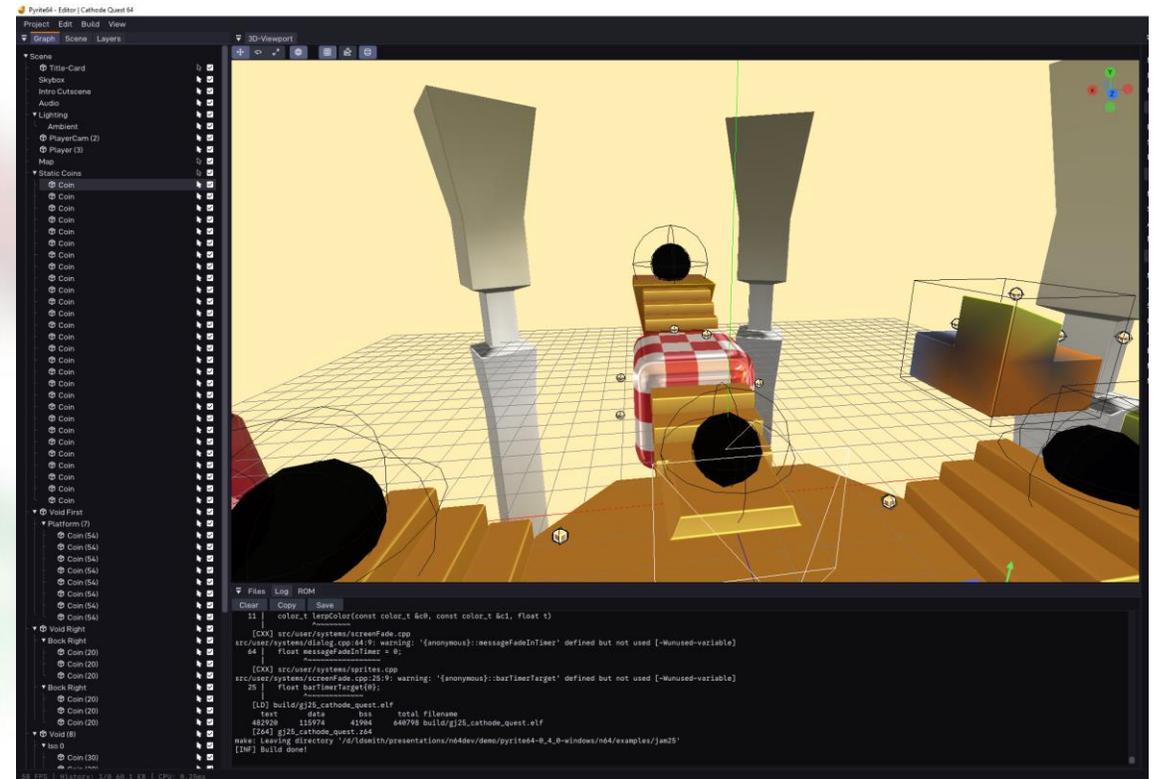
```
409896 108390 40776 559062 build/KnoxGameDesignFirstProject.elf
[Z64] KnoxGameDesignFirstProject.z64
WARNING: Title will be truncated to 20 characters
make: Leaving directory '/d/ldsmith/presentations/pyrite64/projects/first_project/KnoxGameDesignFirstProject'
[INF] Build done!
```



**KNOX**  
**GAME**  
**DESIGN**

# Build a test project

- Open the n64/examples/jam25 example
- Build
- Verify it creates the gj25\_cathode\_quest.z64 file
- Run with gopher64 or ares



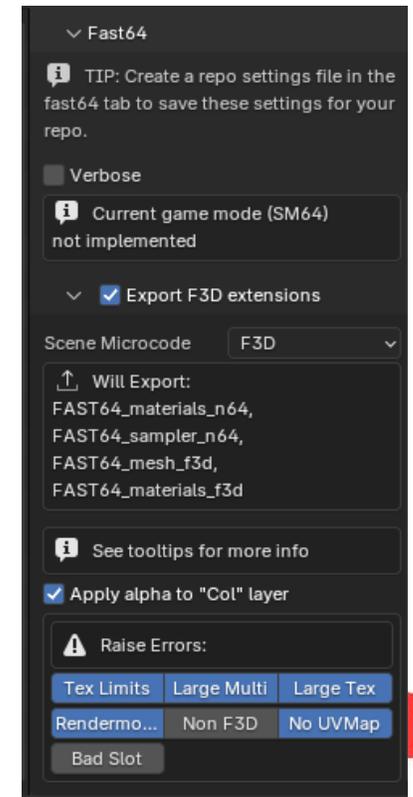
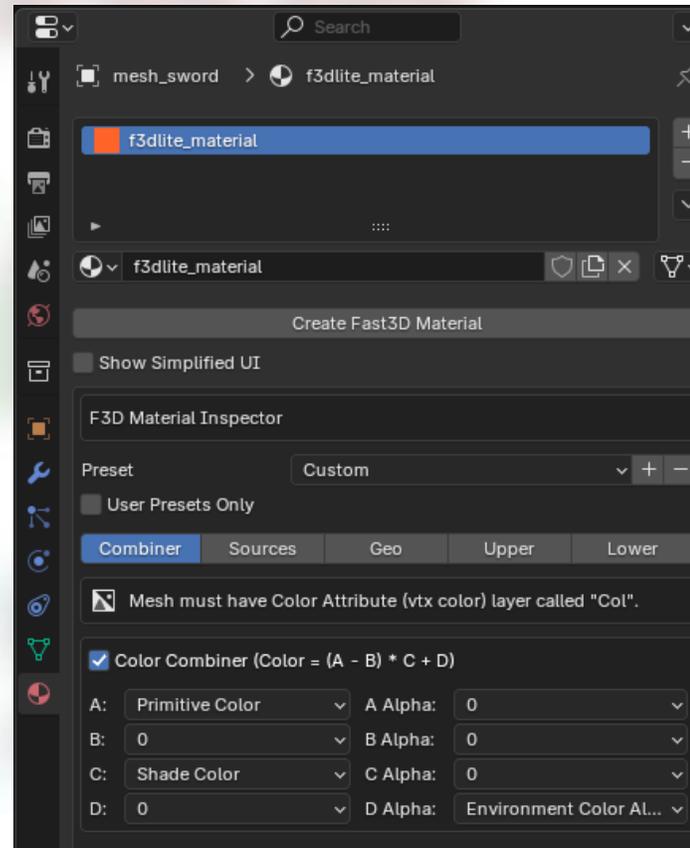
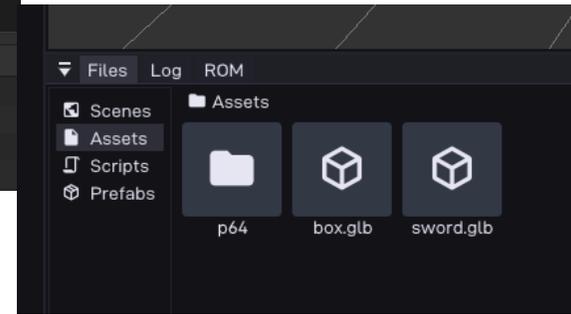
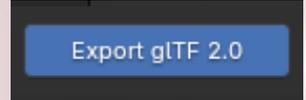
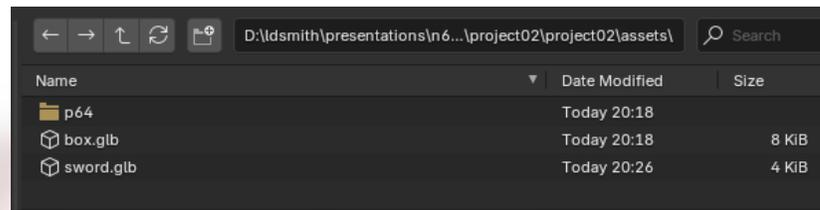
resentations > n64dev > demo > pyrite64-0\_4\_0-windows > n64 > examples > jam25

| Name                   | Date modified     | Type          | Size      |
|------------------------|-------------------|---------------|-----------|
| assets                 | 3/16/2026 9:25 PM | File folder   |           |
| build                  | 3/16/2026 9:26 PM | File folder   |           |
| data                   | 3/1/2026 11:59 PM | File folder   |           |
| engine                 | 3/16/2026 9:26 PM | File folder   |           |
| filesystem             | 3/16/2026 9:26 PM | File folder   |           |
| src                    | 3/16/2026 9:25 PM | File folder   |           |
| tools                  | 3/1/2026 11:59 PM | File folder   |           |
| .gitattributes         | 3/1/2026 11:59 PM | Text Document | 1 KB      |
| .gitignore             | 3/1/2026 11:59 PM | Text Document | 1 KB      |
| CMakeLists.txt         | 3/1/2026 11:59 PM | Text Document | 3 KB      |
| gj25_cathode_quest.z64 | 3/16/2026 9:26 PM | Z64 File      | 18,224 KB |
| Makefile               | 3/16/2026 9:26 PM | File          | 8 KB      |
| Makefile.custom        | 3/1/2026 11:59 PM | CUSTOM File   | 1 KB      |
| project.p64proj        | 3/16/2026 9:25 PM | P64PROJ File  | 1 KB      |



# Creating Blender model

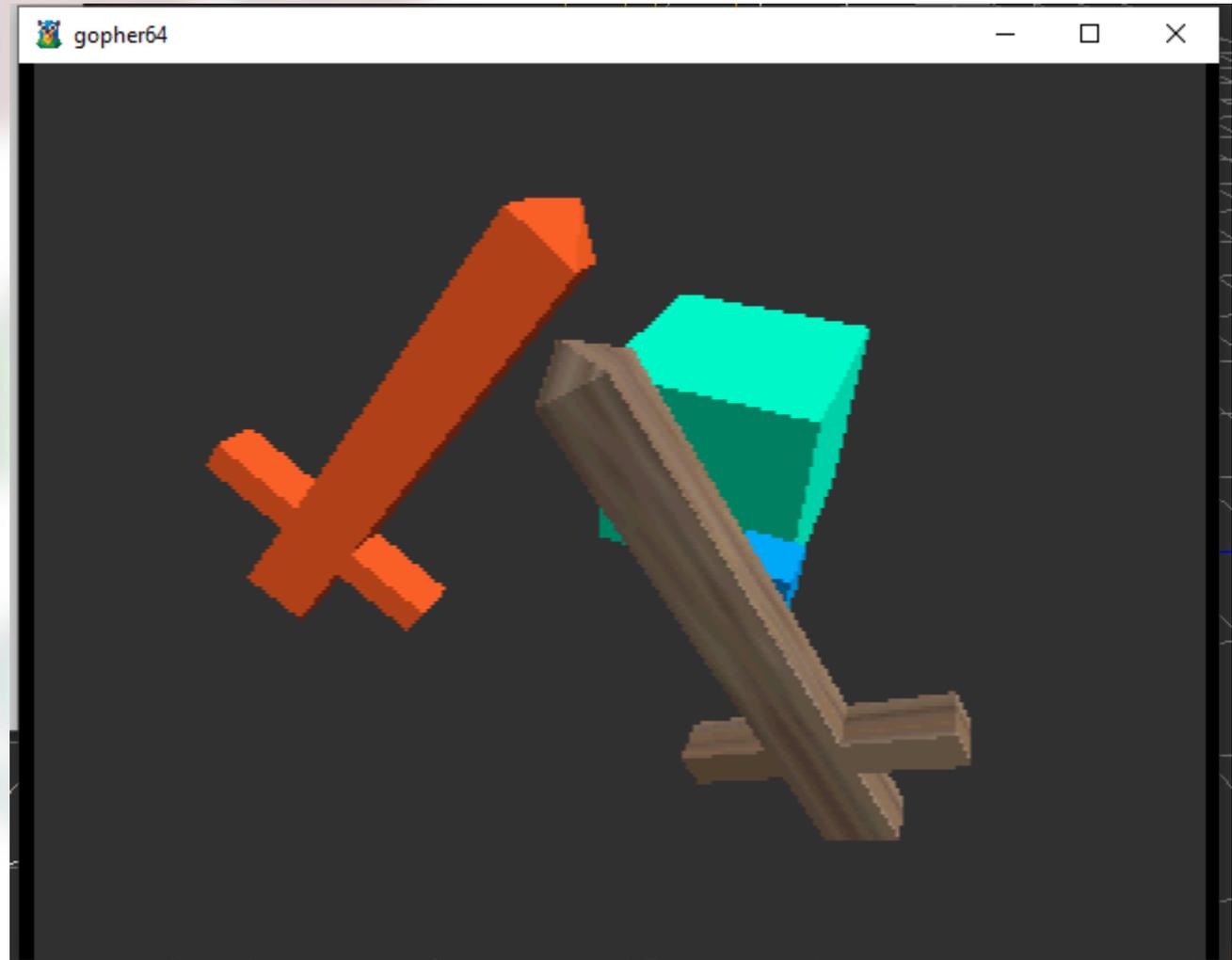
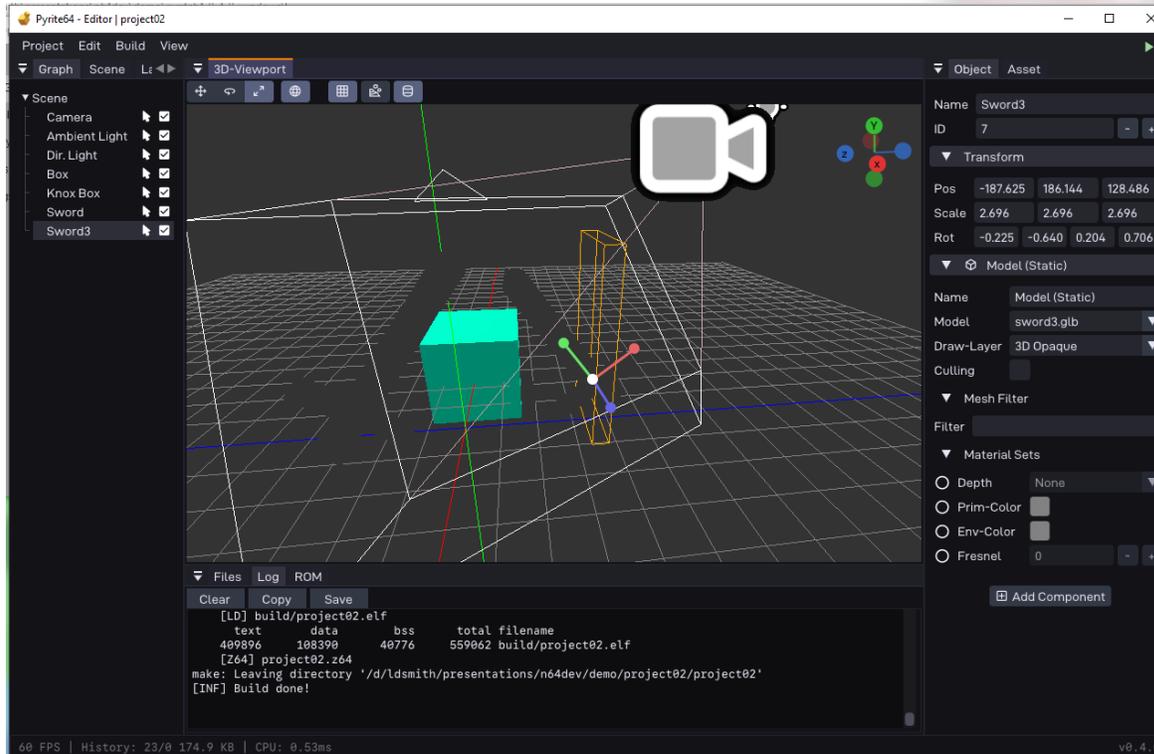
- Create model in Blender
- Must download and enable fast64 addon for Blender
  - <https://github.com/Fast-64/fast64>
  - Edit > Preferences > Addons > Install from disk > Select ZIP file
- File > Export > glTF 2.0
- Make sure to check *Include custom properties on glTF export*
- Copy .glb file to Assets folder
- Will automatically appear in the Pyrite64 Files tab under Assets
- Scene > Add Object, Add Component > Model Static, Model > select .glb file
- Steps for making a Blender model for Pyrite64
  - <https://hailtododongo.github.io/pyrite64/docs/manual/assets/model3d.html>
- If it crashes, the fast64 addon has to be re-enabled in Addons
- If things aren't working, start over from a blank Blender project
- Suggest starting with A: Primitive Color B: 0 C: Shade Color D: 0
- If primitive color works, then try a texture
  - Must be small (such as 32x32 pixels)



```
[ERR] Failed to load 3D model asset: D:\ldsmith\presentations\n64dev\demo\project03\project03\assets\sword.glb -  
Material has no fast64 data! (@TODO: implement fallback)  
If you are using fast64, make sure to enable 'Include -> Custom Properties' during GLTF export
```

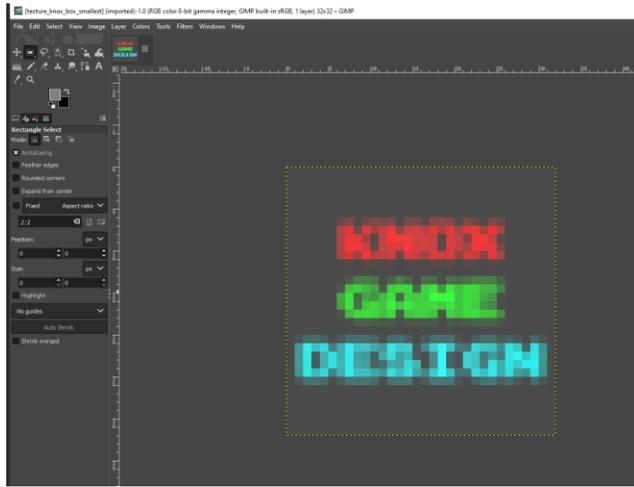


# Blender models imported

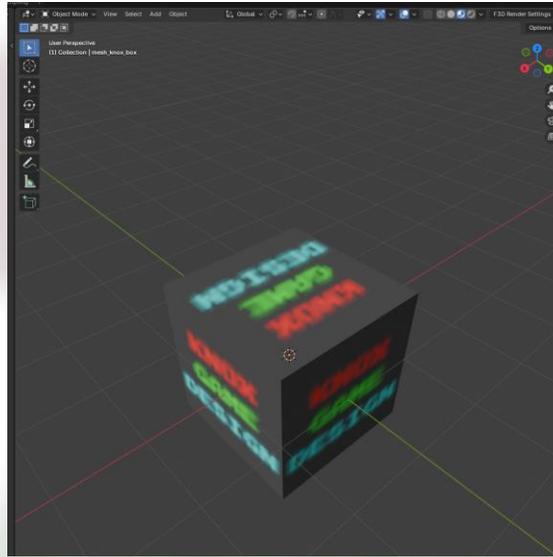


DESIGN

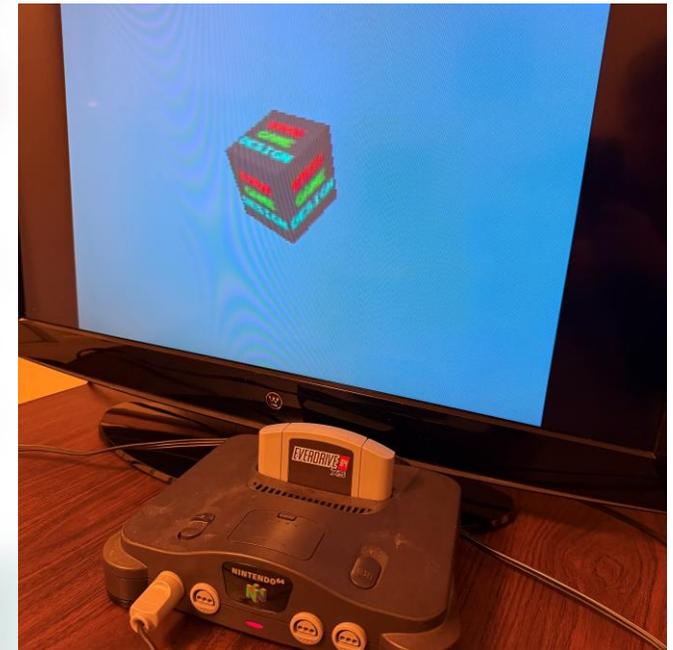
# Texture Mapping



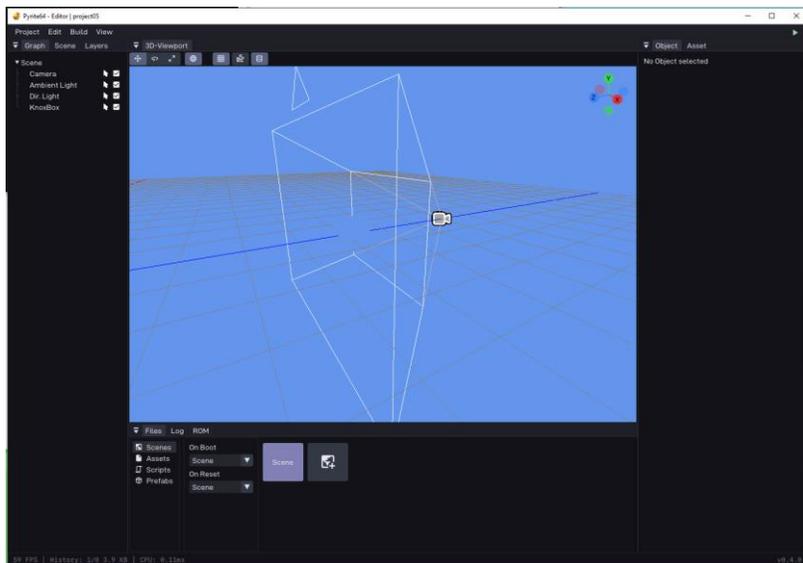
Gimp > .png texture (32x32)



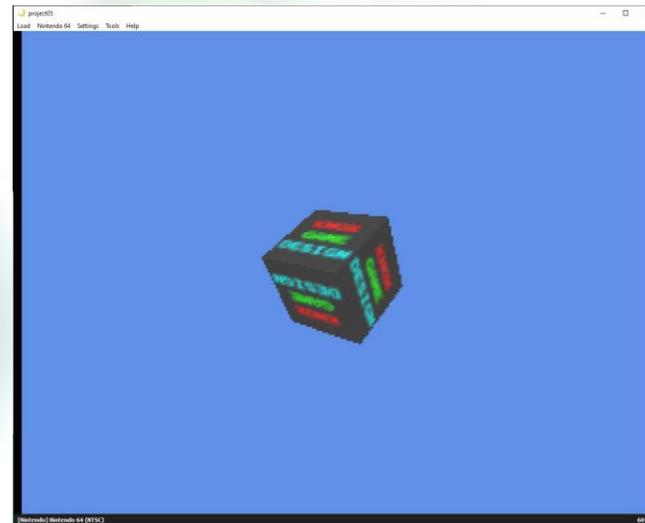
Blender > .glb model



N64 / Everdrive64  
hardware



Pyrite64 > .z64 file

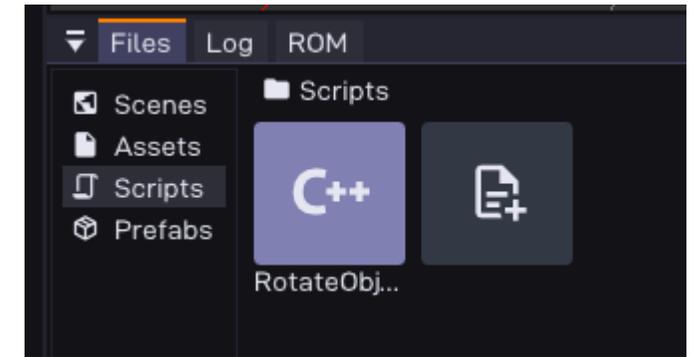
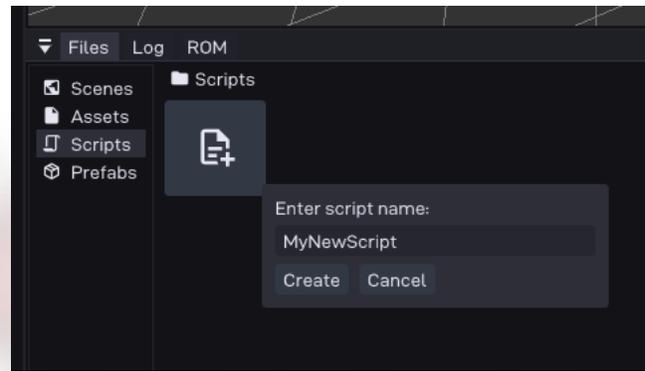


Ares emulator

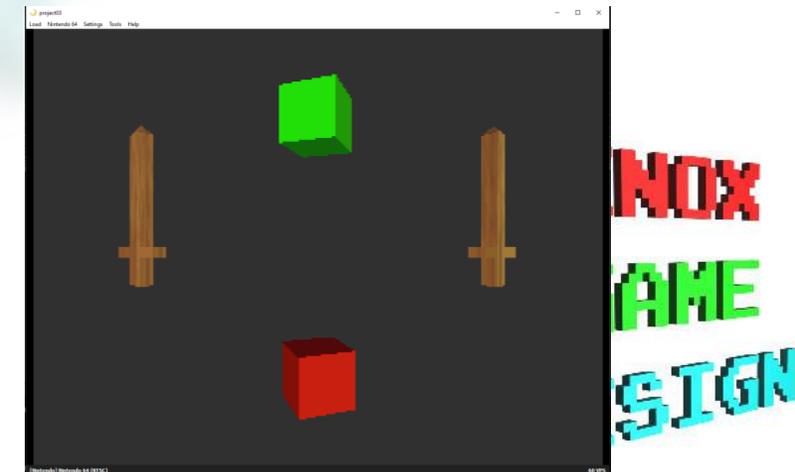
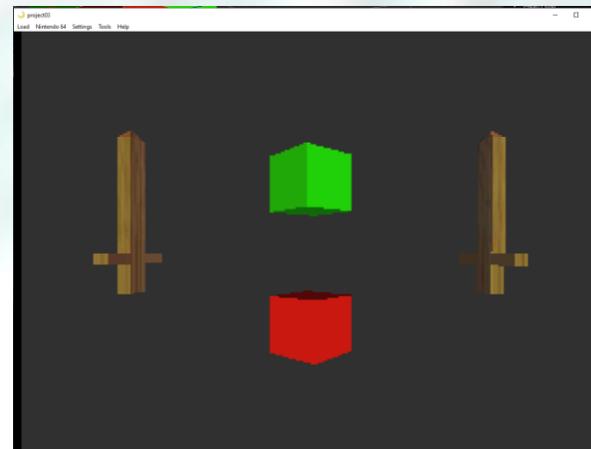
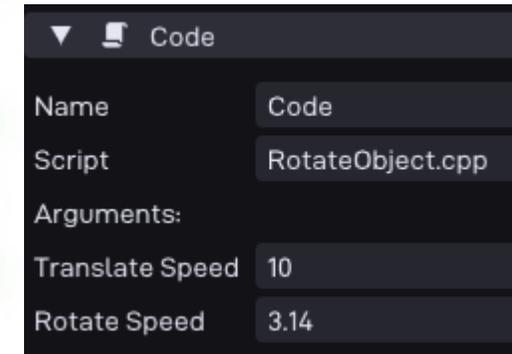
**KNOX**  
**GAME**  
**DESIGN**

# Code

- “scripts” (C++) code is in src/user
- bring your own editor
  - Notepad++, Visual Studio Code, vi, etc
- First create a plain script and attach it to an object
  - Verify that the game still builds and runs
- Can add values to be exposed in the editor
  - Save the .cpp file, and the editor will automatically update
- In Update, add example code to translate and rotate to test
  - Rotation is in radians (3.14 rad = 180 degrees)
- Functions
  - Init
  - Destroy
  - Update
  - Draw
  - OnEvent
  - OnCollision
- Best way to learn is by looking at the examples



```
P64_DATA{  
    [[P64::Name("Translate Speed")]]  
    float translateSpeed;  
  
    [[P64::Name("Rotate Speed")]]  
    float rotateSpeed;  
  
    int noExposed;  
};
```



# API

- Libdragon
- <https://n64squid.com/homebrew/libdragon/>

## Controller control



[Home](#) » [N64 Homebrew](#) » [Libdragon Tutorial](#) » Controller control

```
1 typedef struct __attribute__((packed)) joypad_inputs_s
2 {
3     // Structure containing button presses (see next co
4     joypad_buttons_t btn;
5
6     // Value of the main joystick's X and Y axis
7     // Can range from -128 to +127, but practically is
8     int8_t stick_x;
9     int8_t stick_y;
10
11    // The rest of these are only for GameCube controll
12    // C-stick
13    int8_t cstick_x;
14    int8_t cstick_y;
15    // Analogue triggers
16    uint8_t analog_l;
17    uint8_t analog_r;
18 } joypad_inputs_t;
```

```
1 typedef union joypad_buttons_u
2 {
3     // Raw button data as a 16-bit
4     uint16_t raw;
5     // Button data as callable bits
6     struct __attribute__((packed))
7     {
8         unsigned a : 1;
9         unsigned b : 1;
10        unsigned z : 1;
11        unsigned start : 1;
12        unsigned d_up : 1;
13        unsigned d_down : 1;
14        unsigned d_left : 1;
15        unsigned d_right : 1;
16        unsigned y : 1; // Only for
17        unsigned x : 1; // Only for
18        unsigned l : 1;
19        unsigned r : 1;
20        unsigned c_up : 1;
21        unsigned c_down : 1;
22        unsigned c_left : 1;
23        unsigned c_right : 1;
24    };
25 } joypad_buttons_t;
```

- Tiny3D
- <https://hailtododongo.github.io/tiny3d/annotated.html>

## Tiny3D

[Main Page](#) [Related Pages](#) [Data Structures ▾](#)

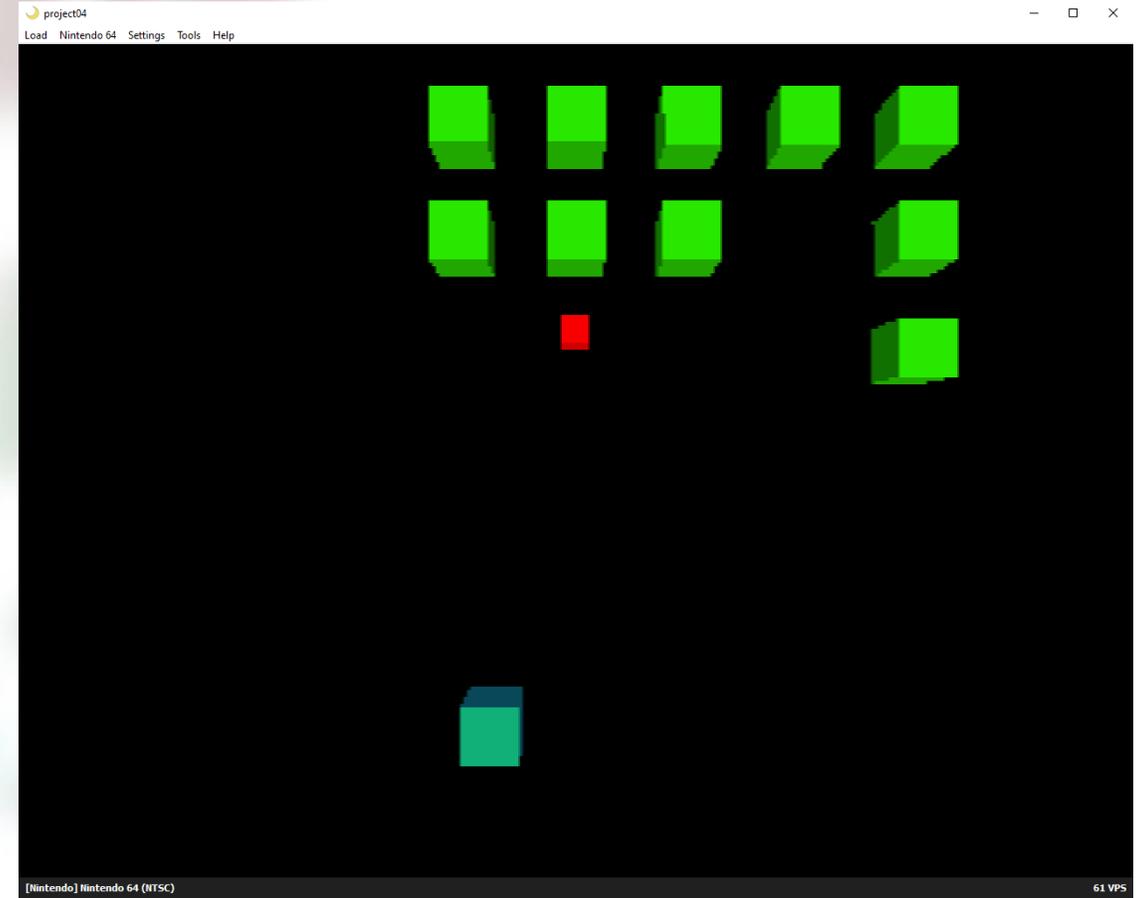
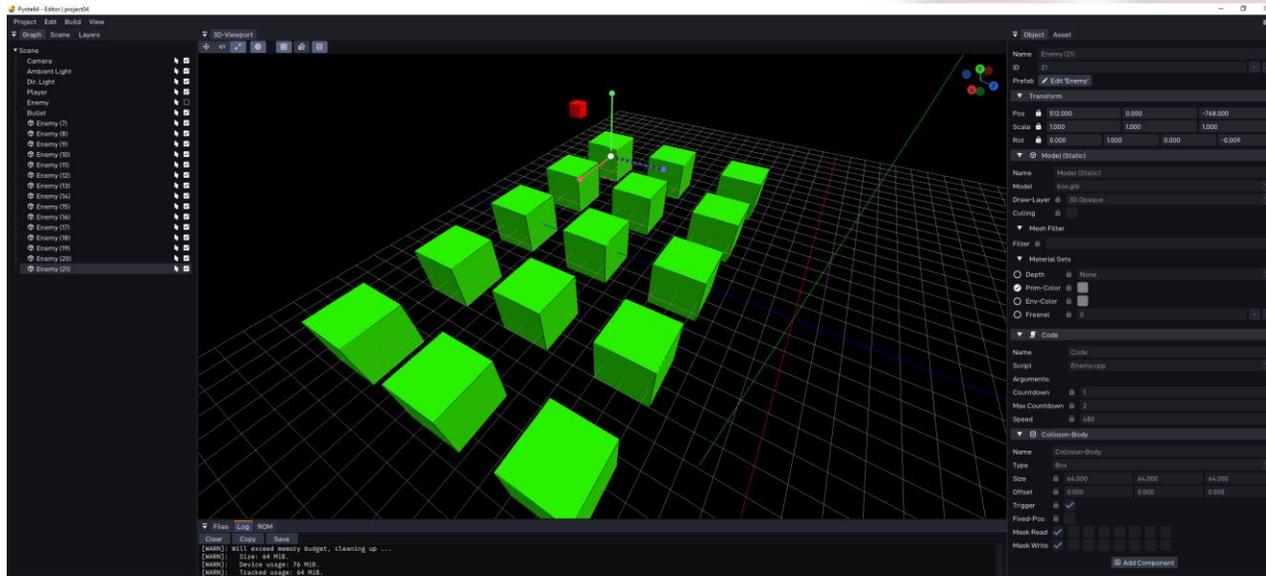
### Data Structures

Here are the data structures with brief descriptions:

|  |  |
|--|--|
| <a href="#">C</a> <code>__attribute__</code>         |  |
| <a href="#">C</a> <code>T3DAnim</code>               |  |
| <a href="#">C</a> <code>T3DAnimChannelMapping</code> |  |
| <a href="#">C</a> <code>T3DAnimKF</code>             |  |
| <a href="#">C</a> <code>T3DAnimTargetBase</code>     |  |
| <a href="#">C</a> <code>T3DAnimTargetQuat</code>     |  |
| <a href="#">C</a> <code>T3DAnimTargetScalar</code>   |  |
| <a href="#">C</a> <code>T3DBone</code>               |  |
| <a href="#">C</a> <code>T3DBvh</code>                |  |
| <a href="#">C</a> <code>T3DBvhData</code>            |  |
| <a href="#">C</a> <code>T3DBvhNode</code>            |  |
| <a href="#">C</a> <code>T3DChunkAnim</code>          |  |
| <a href="#">C</a> <code>T3DChunkBone</code>          |  |
| <a href="#">C</a> <code>T3DChunkOffset</code>        |  |
| <a href="#">C</a> <code>T3DChunkSkeleton</code>      |  |

**KNOX**  
**GAME**  
**DESIGN**

# Collision, Prefabs

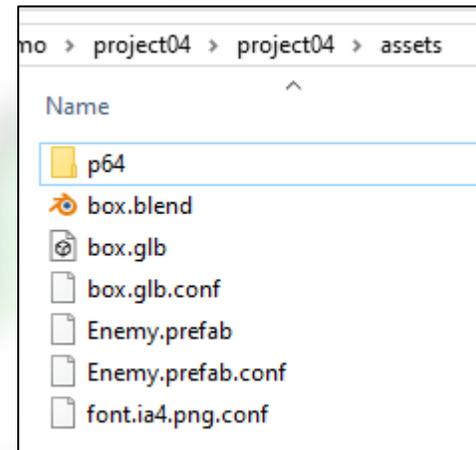
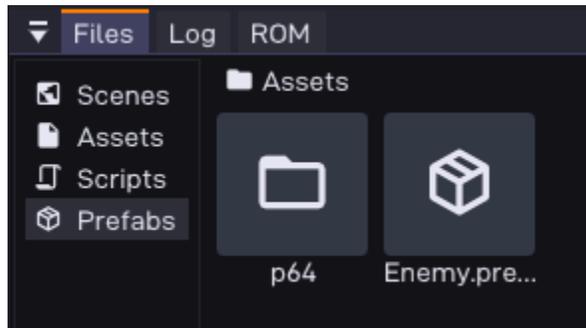


GAME  
DESIGN

Bullet collision with enemy. Does not check collided object type.

```
void onCollision(Object& obj, Data *data, const Coll::CollEvent& event) {  
    obj.remove();  
}
```

Enemy Prefab: All objects of this type will inherit the prefab properties. Located in the assets folder.



Shooting: get reference to bullet object from player object. Set bullet position to player position

```
if (joypad.btn.a) {  
    P64::Object *objBullet = SceneManager::getCurrent().getObjectById(6);  
    objBullet->pos.x = obj.pos.x;  
    objBullet->pos.z = obj.pos.z;  
}
```

**KNOX**  
**GAME**  
**DESIGN**

# Playing your game on hardware

- Cartridge

- EverDrive64 X5
  - krikzz.com
- EverGenesis 64
  - Kamisama Games

- System

- Nintendo 64
  - Nintendo
- Analogue<sup>3D</sup>
  - analogue.co



**KNOX**  
**GAME**  
**DESIGN**

# Resources

- N64 Brew
  - [https://n64brew.dev/wiki/Main\\_Page](https://n64brew.dev/wiki/Main_Page)
- Awesome N64 Development
  - <https://n64.dev/>
- Pyrite<sup>64</sup>
  - <https://hailtododongo.github.io/pyrite64/index.html>
- libdragon
  - <https://github.com/DragonMinded/libdragon>
- tiny3d
  - <https://github.com/HailToDodongo/tiny3d>
- gopher64
  - <https://github.com/gopher64/gopher64/releases/tag/v1.1.15>