Quero desenvolver um projeto de um boneco 3d que pode mover-se em um espaço 3d, como um diorama. Como faço isso usando Blender 3d para Nintendo DS? Preferencialmente usando frameworks de teste em C?

Criar um **boneco 3D animado e interativo** para o **Nintendo DS**, usando o **Blender** e **C com frameworks de teste**, é uma tarefa desafiadora mas super legal! Vamos quebrar esse projeto em partes pra você ter um caminho claro:

**🧩 Etapas do Projeto**

**1. Modelagem no Blender (para Nintendo DS)**

O Nintendo DS tem limitações sérias de hardware, então:

* Use o **Blender** para modelar seu boneco em **low poly** (baixa contagem de vértices/polígonos).
* **Texturas** devem ser pequenas (por exemplo, 128x128 ou 64x64).
* Use **UV Mapping** simples.
* Use **apenas 1 material por objeto**.
* Formatos recomendados para exportação:
  + .obj (com UV)
  + .imdl ou .nsbmd (formatos internos da Nintendo, conversão necessária)

**2. Exportar e Converter Modelos para o Nintendo DS**

Blender não exporta diretamente para o DS. Você precisa:

**🔧 Ferramentas recomendadas:**

* **MKDS Course Modifier** (para conversão)
* **NSMBe** (para hack de ROMs, se for usar jogo base)
* **devkitPro** com **libnds** (ambiente de desenvolvimento oficial homebrew)
* **G3D converter (g3dcvtr.exe)** – da Nintendo SDK (você pode usar o substituto open-source se não tiver acesso)

**Caminho típico:**

Blender → Exportar .obj/.dae → Converter para .nsbmd com g3dcvtr ou tools open-source → Usar no DS

**🧠 3. Programação no Nintendo DS (em C)**

Você precisa instalar:

**🧰 Ferramentas:**

* [devkitPro](https://devkitpro.org/)
* [libnds](https://github.com/devkitPro/libnds)
* [nds-examples](https://github.com/devkitPro/nds-examples)

Instale com:

sudo dkp-pacman -S nds-dev

**🛠️ Estrutura básica:**

Você cria um loop principal (main) com lógica de entrada, atualização e renderização.

int main(void) {

consoleDemoInit(); // inicializa texto na tela

while (1) {

scanKeys();

int keys = keysHeld();

if (keys & KEY\_UP) { /\* mover boneco \*/ }

swiWaitForVBlank();

}

return 0;

}

**🎮 4. Mover o boneco em um espaço 3D (Diorama)**

No DS, você tem **hardware 3D limitado (modo 0)**.

Você pode usar *gl functions da libnds*\* para desenhar e mover o boneco:

glLoadIdentity();

glTranslatef32(posX, posY, posZ);

glRotatef32(angle, 0, 1<<12, 0);

glBindTexture(0, textureID);

glCallList(modelID);

**🧪 5. Testes com Framework em C**

Nintendo DS não suporta frameworks como GoogleTest, mas você pode:

* Usar **Unity Test Framework** (não relacionado à Unity 3D!): [Unity Test](https://github.com/ThrowTheSwitch/Unity)
* Criar funções simples de assertiva em C para testes manuais:

void test\_movement() {

assert(move\_player(0,1) == expectedY);

}

**🧠 Dicas finais**

* Use **emulador DeSmuME** ou **melonDS** para testar rapidamente.
* Otimize seu modelo 3D — o DS suporta até 2048 polígonos por frame aprox.
* Use texturas paletizadas (.tex/.tpl) para economizar VRAM.
* Se quiser animações, você vai precisar “bakear” quadros ou usar bones via exportadores especializados.

Se quiser, posso te montar um **projeto base** com:

* Makefile
* Código base em C para movimento
* Modelo 3D de exemplo pronto para uso no DS

Ou até te ajudar a configurar seu Blender pra exportar corretamente. Como quer seguir?

Montar projeto base

Show! Vamos montar um **projeto base** completo pra você começar a desenvolver seu boneco 3D interativo no Nintendo DS, com o Blender como ferramenta de modelagem e C com libnds para programar.

## 📦 Estrutura do Projeto

ds-boneco3d/

├── Makefile

├── source/

│ ├── main.c

│ ├── model.h

│ └── model.c

├── include/

│ └── model.h

├── gfx/

│ ├── boneco.imd (modelo exportado)

│ └── boneco.nsbmd (modelo DS)

├── build/

└── README.md

## ⚙️ Makefile (base para devkitPro + libnds)

ARM7BIN := $(LIBNDS)/default\_arm7.bin

include $(DEVKITARM)/ds\_rules

TARGET := boneco3d

BUILD := build

SOURCE := source

DATA := gfx

INCLUDE := include

LIBS := -lfat -lnds9

OBJS := $(BUILD)/main.o $(BUILD)/model.o

all: $(BUILD) $(TARGET).nds

$(BUILD):

mkdir -p $(BUILD)

$(BUILD)/%.o: $(SOURCE)/%.c

@echo Compiling $<

$(CC) -MMD -MP -MF $(@:.o=.d) -I$(INCLUDE) -c $< -o $@

$(TARGET).nds: $(OBJS)

@echo Linking...

$(CC) $(OBJS) -o $@ $(LIBS)

clean:

rm -rf $(BUILD) \*.nds

-include $(OBJS:.o=.d)

## 🧠 main.c – loop principal e controle

#include <nds.h>

#include <gl2d.h>

#include "model.h"

int main(void) {

videoSetMode(MODE\_0\_3D);

vramSetBankA(VRAM\_A\_TEXTURE);

glInit();

glEnable(GL\_TEXTURE\_2D);

glClearColor(0, 0, 0, 31);

glClearPolyID(63);

glClearDepth(0x7FFF);

modelInit(); // inicializa modelo 3D

float x = 0, z = -5;

float angle = 0;

while (1) {

scanKeys();

int keys = keysHeld();

if (keys & KEY\_UP) z += 0.1f;

if (keys & KEY\_DOWN) z -= 0.1f;

if (keys & KEY\_LEFT) x -= 0.1f;

if (keys & KEY\_RIGHT) x += 0.1f;

glPushMatrix();

glLoadIdentity();

glTranslatef(x, 0.0f, z);

glRotatef(angle, 0, 1, 0);

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

drawModel();

glFlush(0);

glPopMatrix();

swiWaitForVBlank();

}

return 0;

}

## 📁 model.c e model.h (mock para uso com .nsbmd)

// model.h

#pragma once

void modelInit();

void drawModel();

// model.c

#include <nds.h>

#include "model.h"

void modelInit() {

// Aqui você pode carregar o .nsbmd e .nsbtx

// usando o sistema de arquivos se quiser.

}

void drawModel() {

// Placeholder para desenhar um cubo (ou seu modelo futuramente)

glBegin(GL\_QUADS);

glColor3b(255, 0, 0);

glVertex3f(-1, -1, 0);

glVertex3f(1, -1, 0);

glVertex3f(1, 1, 0);

glVertex3f(-1, 1, 0);

glEnd();

}

## 📦 gfx/ – Modelos 3D

Você vai colocar aqui o modelo exportado do Blender e convertido com g3dcvtr ou bmdconv.

* Exporte seu modelo como .obj ou .dae.
* Use g3dcvtr.exe para converter:

g3dcvtr boneco.imd

g3dcvtr -n boneco.imd

## 🧪 Extras – Teste de função (simples)

Se quiser testar a movimentação em C:

// Em model.c

float movePlayer(float current, int direction) {

return current + (direction \* 0.1f);

}

// Em main ou test

assert(movePlayer(1.0f, 1) == 1.1f);

## 🕹 Como rodar

1. Instale [devkitPro + libnds](https://devkitpro.org/wiki/Getting_Started)
2. Compile com:

make

1. Teste com emulador **melonDS** ou **DeSmuME**
2. Ou use um **flashcard** para rodar no DS real

## ⚡ Quer o ZIP com esse projeto pronto?

Posso gerar pra você um .zip com tudo isso montado, incluindo um exemplo de cubo animado, ou até com um modelo 3D simples feito no Blender. Quer que eu te mande esse pacote base? Se sim, me diz:

* 💻 Qual sistema você está usando? (Windows/Linux/macOS)
* 📦 Quer que o exemplo inclua modelo 3D .obj ou já convertido?
* 🧊 Quer um boneco simples de exemplo ou um cubo por enquanto?

Aí te entrego o projeto redondinho 😎