

# Pipeline 3D $\longrightarrow$ 2D

Modelo 3D a Pixel Art

Autor:	Martín Molina
Profesor:	Patricio Inostroza
Auxiliares:	Elías Moreno Vergara Francisco Ortiz
Ayudantes:	Javiera Labrín A. Diego Navarrete Gonzalez

# Índice de Contenidos

<b>1. Herramientas utilizadas</b>	<b>1</b>
1.1. Godot . . . . .	1
1.2. The Models Resource . . . . .	1
1.3. Firefox . . . . .	1
<b>2. Objetivos del proyecto</b>	<b>2</b>
2.1. Objetivos Propuestos . . . . .	2
2.2. Objetivos Logrados . . . . .	2
2.3. Objetivos No Logrados . . . . .	2
2.4. Adicional . . . . .	2
<b>3. Pantallazos</b>	<b>3</b>
3.1. Interfaz del programa . . . . .	3
3.2. Cargar un modelo 3D . . . . .	3
3.3. Efecto Pixel Art . . . . .	4
3.4. Efecto Pixel Art + Efecto CRT . . . . .	4

# Índice de Figuras

1. Interfaz del programa sin cargar ningun modelo 3D. . . . .	3
2. Programa con un modelo 3D cargado sin efectos. . . . .	3
3. Programa con un modelo 3D con el efecto Pixel Art aplicado. . . . .	4
4. Programa con un modelo 3D con el efecto Pixel Art y el efecto de CRT aplicado. . . . .	4

# 1. Herramientas utilizadas

## 1.1. Godot

“Godot Engine es un motor de juegos multiplataforma repleto de funciones para crear juegos en 2D y 3D desde una interfaz unificada. Ofrece un completo conjunto de herramientas comunes, para que los usuarios puedan centrarse en crear juegos sin tener que reinventar la rueda. Los juegos se pueden exportar con un solo clic a varias plataformas, incluidas las principales plataformas de escritorio (Linux, macOS, Windows), plataformas móviles (Android, iOS), así como plataformas basadas en web y consolas.” (Extraído de su pagina en GitHub)

Godot fue el único programa que utilicé para hacer mi proyecto, ya que viene muy bien integrado con todo lo que necesitaba, tiene su propio lenguaje para shaders y scripts, una interfaz muy simple para alguien que está recién aprendiendo, pero la verdadera razón por la que escogí a Godot es por su excelente soporte en Linux (sistema operativo que utilizo principalmente).

Fue una experiencia muy grata utilizar este programa aunque si bien hay muchísimos más tutoriales para Unity fue sencillo encontrar todo lo que necesitaba.

## 1.2. The Models Resource

The Models Resource fue la pagina web principal de donde conseguía los modelos para mi programa, y basicamente consiste en una web donde un grupo de aficionados van recompilando modelos de videojuegos tanto antiguos como nuevos y los hacen públicos.

Mi programa solo ha sido probado con modelos de este sitio web.

## 1.3. Firefox

Y bueno obviamente utilicé un navegador para buscar recursos, documentación y tutoriales.

## 2. Objetivos del proyecto

### 2.1. Objetivos Propuestos

- Aprender a utilizar algun programa relacionado a la grafica (Godot o Unity).
- Desarrollar un pipeline que lleve un modelo 3D a Pixel Art de manera simple y sencilla.
- Lograr tambien limitar las paletas de colores de los modelos 3D.
- Lograr aplicar paletas de colores limitadas para así emular aún mejor el efecto de consolas antiguas.

### 2.2. Objetivos Logrados

- Aprendí a utilizar de manera cómoda Godot 4.
- Aprendí como funcionan los shaders de Godot.
- Logré hacer un programa que recibe un modelo 3D y lo transforma a Pixel Art.
- Logré limitar la paleta de color de cada modelo para lograr un efecto realista.

### 2.3. Objetivos No Logrados

- No logré aplicar paletas de colores personalizadas para cada modelo, pues se me complicó la programación del shader.

### 2.4. Adicional

- Logré aplicar un shader que emula el efecto de las pantallas CRT.
- Decoré un poco el programa con una font bonita.

## 3. Pantallazos

### 3.1. Interfaz del programa

Allí se muestra los efectos que pude lograr y el boton para cargar un modelo 3D.



Figura 1: Interfaz del programa sin cargar ningun modelo 3D.

### 3.2. Cargar un modelo 3D



Figura 2: Programa con un modelo 3D cargado sin efectos.

### 3.3. Efecto Pixel Art



Figura 3: Programa con un modelo 3D con el efecto Pixel Art aplicado.

### 3.4. Efecto Pixel Art + Efecto CRT



Figura 4: Programa con un modelo 3D con el efecto Pixel Art y el efecto de CRT aplicado.