**EXAMEN FINAL COM 300 – INTELIGENCIA ARTIFICIAL**

ESTUDIANTE: Jorge Andrés Soruco Ibáñez

**TITULO DEL PROYECTO**

generación de trabajos procedurales inteligentes con clustering de texturas

**1. Introducción**

En este proyecto se desarrolla un sistema inteligente de asignación de texturas en Blender utilizando técnicas de *clustering* no supervisado. Se busca automatizar el flujo de trabajo en escenas 3D mediante la clasificación previa de texturas y su asignación inteligente a objetos modelados, mejorando así la eficiencia creativa y la lógica visual de los entornos generados.

**2. Objetivos**

* Implementar un algoritmo de aprendizaje no supervisado para agrupar texturas por similitud visual.
* Integrar los resultados del clustering en Blender mediante scripts en Python.
* Diseñar una interfaz en Blender que permita aplicar texturas de forma contextual y automática.
* Documentar el flujo de trabajo completo desde la recolección de datos hasta la visualización final

**3. Recursos empleados**

* **Dataset:**  
  Se utilizaron 120 imágenes del tipo “cracked” extraídas del *Describable Textures Dataset (DTD)*, así como texturas adicionales provenientes de sitios como ambientCG y CC0Textures.
* **Plataformas y herramientas:**
* Google Colab: entrenamiento del modelo de clustering con K-means.
* Blender (v. [especificar]): motor 3D para integración y visualización.
* Python (con bpy, scikit-learn, PIL, json, numpy)
* GitHub: repositorio del proyecto.

**4. Metodología**

4.1 Preprocesamiento y análisis no supervisado

Las texturas fueron procesadas en Google Colab extrayendo su color promedio como vector de características. Luego, se aplicó el algoritmo *K-means* con 4 a 47 clusters, según el experimento, para obtener agrupaciones visualmente coherentes.

from sklearn.cluster import KMeans

modelo = KMeans(n\_clusters=4)

clusters = modelo.fit\_predict(caracteristicas)

Los resultados fueron exportados a un archivo clusters\_texturas.json con el formato:

{

"textura\_01.jpg": 2,

"textura\_02.jpg": 0

}

4.2 Integración con Blender

Dentro de Blender, se diseñó un script en Python que:

* Lee el archivo JSON.
* Crea materiales por grupo de clustering.
* Aplica automáticamente una textura adecuada al objeto activo.
* Ofrece un **panel interactivo personalizado** para aplicar texturas inteligentes desde la interfaz.

4.3 Casos de prueba

Se generó una “galería visual” con planos que muestran todas las texturas tipo *cracked*, y se realizaron pruebas aplicando materiales a objetos con distintas formas (planos, sillas, cubos) para verificar que el sistema asignara materiales coherentes visualmente

**5. Resultados**

* Se logró una asignación automática de texturas basada en agrupaciones cromáticas, detectadas sin supervisión.
* La integración del panel personalizado permitió aplicar materiales con un solo clic, según la lógica de clustering.
* El proyecto demostró cómo la IA puede apoyar flujos creativos en diseño y modelado 3D, con lógica contextual.

**6. Conclusiones**

Este proyecto demuestra la viabilidad de usar inteligencia artificial no supervisada para generar entornos 3D más eficientes y visualmente coherentes. La combinación de *machine learning*, automatización con Python y entornos gráficos 3D potencia notablemente la creación artística y técnica.

A futuro, se pueden incorporar modelos más complejos (como redes neuronales convolucionales o clasificadores de estilo) o extender el sistema para detectar automáticamente tipos de objetos (mesas, paredes, techos) y sugerir materiales aún más específicos.