



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e
INTERACCIÓN HUMANO COMPUTADORA



REPORTE DE PRÁCTICA N° 06

NOMBRE COMPLETO: Lopez Flores Diego Alberto

N° de Cuenta: 315081143

GRUPO DE LABORATORIO: 11

GRUPO DE TEORÍA: 04

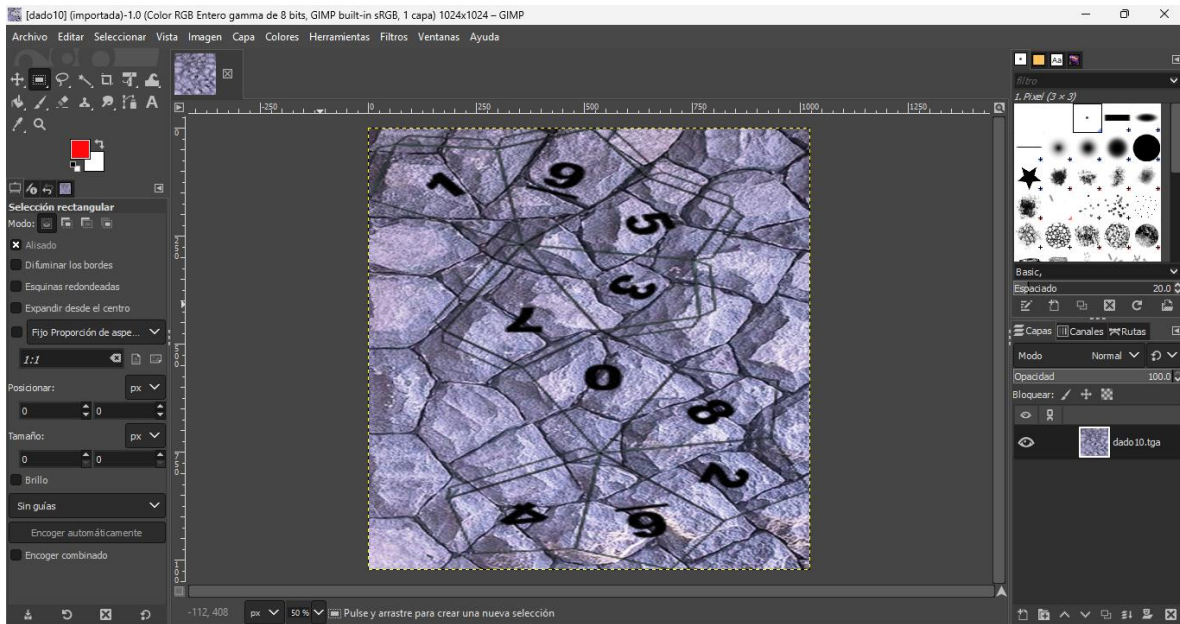
SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: 25 de septiembre de 2024

CALIFICACIÓN: _____

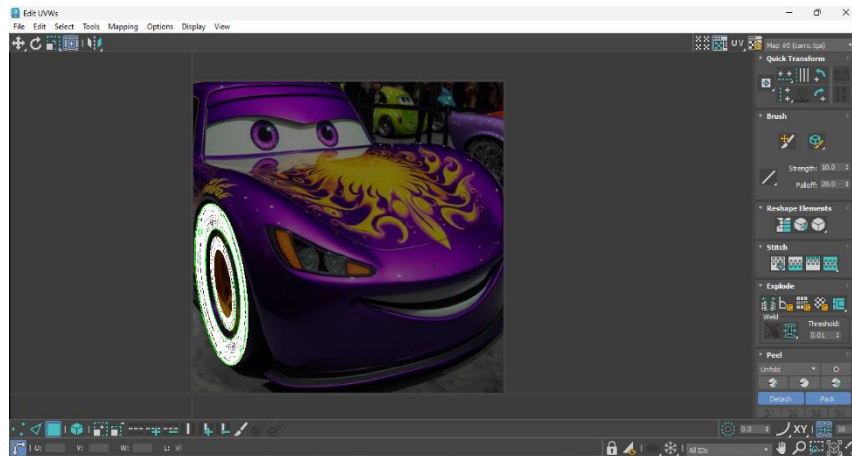
Ejercicio 1: Crear un dado de 10 caras y texturizarlo por medio de código

Utilizando el software para edición de imagen Gimp se obtuvo una plantilla y se personalizo para usarla en la textura del dado

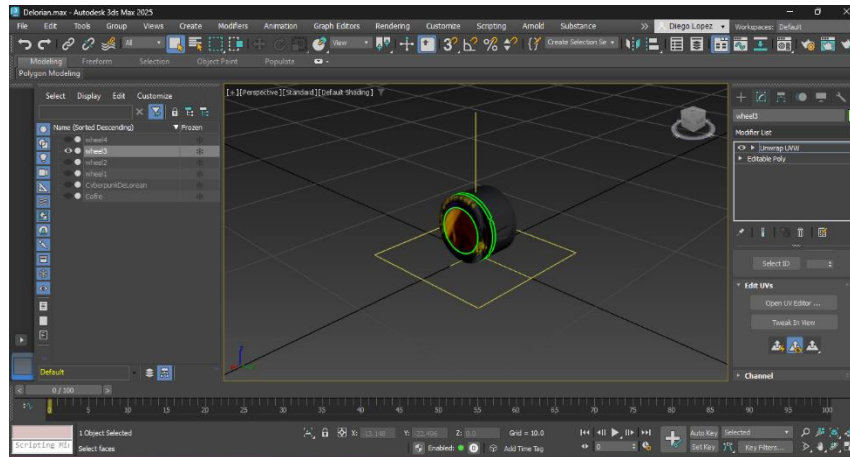


Ejercicio 2: Importar el modelo de su coche con sus 4 llantas acomodadas y tener texturizadas las 4 llantas (diferenciar caucho y rin)

Usando el texturizado de mipmap en 3dsMax separamos las 4 ruedas y después se le aplica textura usando la imagen que tenemos como referencia



Una vez texturizada la llanta ahora repetimos el procedimiento con las 3 llantas restantes



Por ultimo lo cargamos en openGL , creamos las llantas tipo modelo

```
Model Wheel1_M;
Model Wheel2_M;
Model Wheel3_M;
Model Wheel4_M;
```

Y las cargamos con su ruta

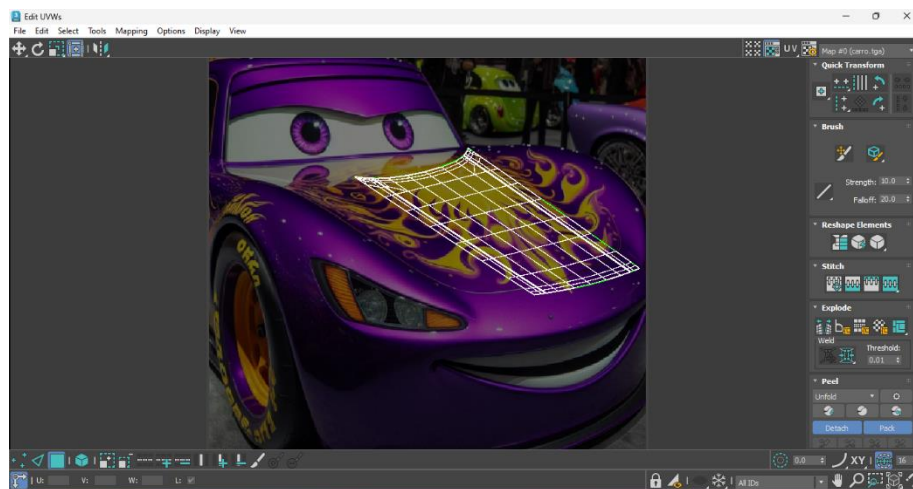
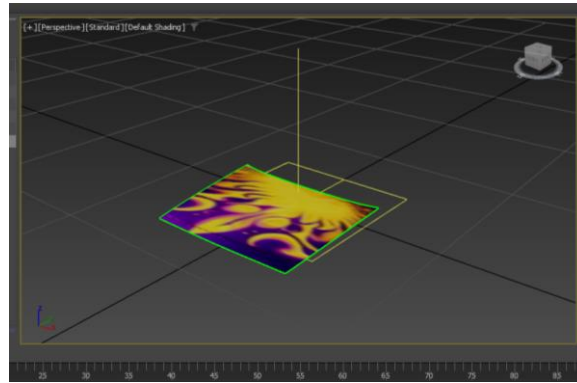
```
Wheel1_M = Model();
Wheel1_M.LoadModel("Models/Wheel1.obj");
Wheel2_M = Model();
Wheel2_M.LoadModel("Models/Wheel2.obj");
Wheel3_M = Model();
Wheel3_M.LoadModel("Models/Wheel3.obj");
Wheel4_M = Model();
Wheel4_M.LoadModel("Models/Wheel4.obj");
```

Las dibujamos con el render model, al usar el modelado en jerarquía solo cargamos la modelAux donde tenemos la base de carro

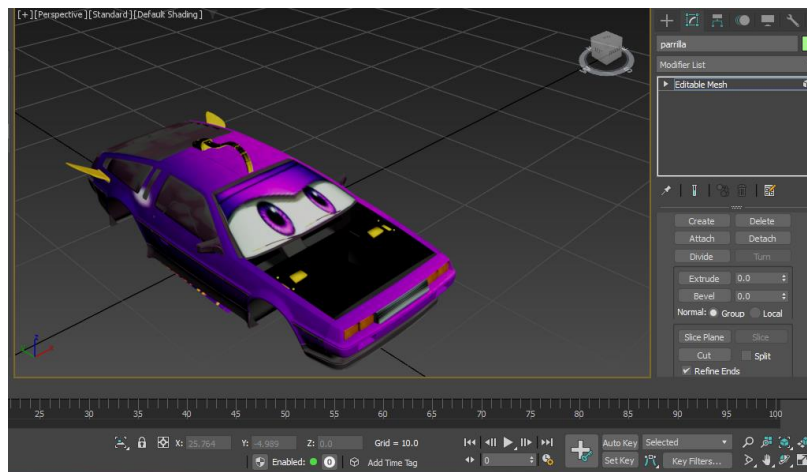
```
//Llanta delantera izquierda
model = modelaux;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Wheel2_M.RenderModel();
```

Ejercicio 3: Texturizar la cara del personaje de la imagen tipo cars en el espejo (ojos) y detalles en cofre y parrilla de su propio modelo de coche

Para este ejercicio separamos momentáneamente el cofre, la parrilla y el parabrisas ya que el mesh del delorian estaba revuelta en la UVW, usando la misma imagen y acomodando la textura queda



El resto del delorian se texturizo de un color igual tomado de la referencia



Exportamos como obj, y cargamos los obj en el openGL, creando un nuevo modelo y llamándolo a cargar

```
Model Cofre_M;
Model Delorian_M;
```



```

Delorian_M = Model();
Delorian_M.LoadModel("Models/Delorian.obj");
Cofre_M = Model();
Cofre_M.LoadModel("Models/Cofre.obj");

```

Instanciamos al coche y lo ubicamos donde queremos, como este será nuestro nodo padre todas las rotaciones, traslaciones y escalamientos tiene que heredarse para evitar problemas

```

//Instancia del coche
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + mainWindow.getmuevex() , -0.5f,
-3.0f));
modelaux = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Delorian_M.RenderModel();

```



Conclusión

En esta práctica, se aplicaron los conocimientos adquiridos sobre texturizado en gráficos 3D, implementando el proceso completo de aplicación de texturas en un modelo de carro. Se utilizó **3dsMax** como software principal para modelado y texturizado, donde se incorporaron técnicas como el uso de **mipmaps** para mejorar el rendimiento visual y la calidad de las texturas a diferentes distancias de visualización. Además, se empleó una imagen de referencia en formato **.tga**, lo que permitió trabajar con un formato de alta calidad, que admite canales de transparencia y ofrece una excelente representación de color.