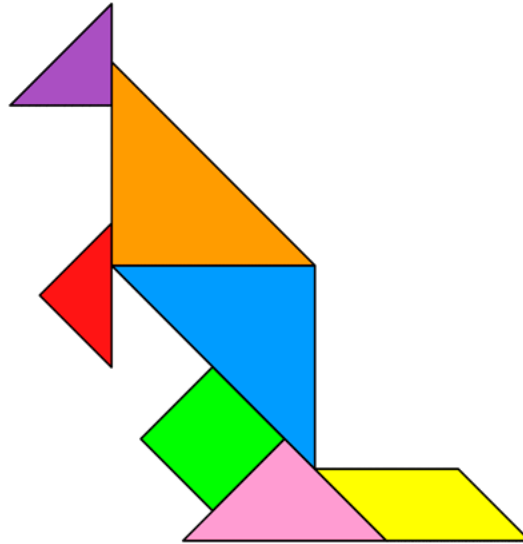


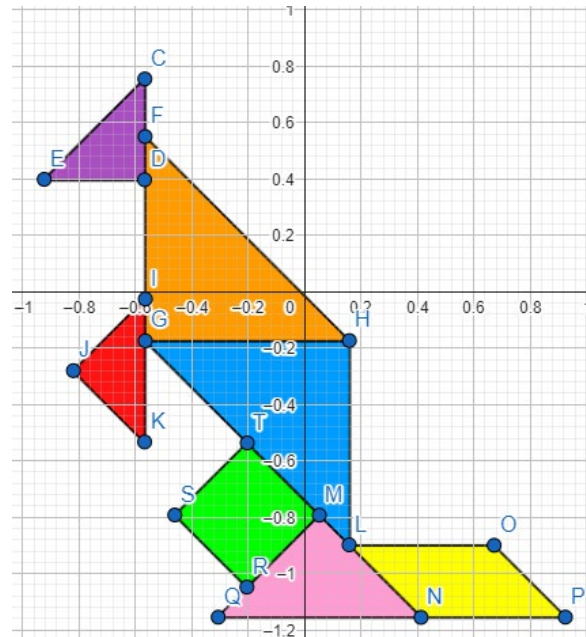
Práctica 2

Desarrollo.

1. Imagen de referencia para la generación de un canguro mediante primitivas 2D y figuras geométricas.



2. Ajustamos la imagen del canguro para que esté en ubicación dentro de un cuadrado con lado igual a 2 y centrado en el origen del plano cartesiano. Colocamos los vértices correspondientes a cada figura.



3. Generamos los puntos en un Excel para que sea más sencillo colocarlos en el proyecto de Visual Studio.

V	x	y	Linea Visual Studio
C=	-0.571514379	0.943139226	-0.571514379f, 0.943139226f, 0.00f, 1.00f, 0.00f, 0.00f, //C=
D=	-0.572411733	0.585448842	-0.572411733f, 0.585448842f, 0.00f, 1.00f, 0.00f, 0.00f, //D=
E=	-0.928394144	0.586944576	-0.928394144f, 0.586944576f, 0.00f, 1.00f, 0.00f, 0.00f, //E=
F=	-0.570733978	0.738844240	-0.570733978f, 0.73884424f, 0.00f, 1.00f, 0.00f, 0.00f, //F=
G=	-0.569932484	0.013836890	-0.569932484f, 0.01383689f, 0.00f, 1.00f, 0.00f, 0.00f, //G=
H=	0.153827133	0.014666212	0.153827133f, 0.014666212f, 0.00f, 1.00f, 0.00f, 0.00f, //H=
I=	-0.569593663	0.162211236	-0.569593663f, 0.162211236f, 0.00f, 1.00f, 0.00f, 0.00f, //I=
J=	-0.824822469	-0.091923715	-0.824822469f, -0.091923715f, 0.00f, 1.00f, 0.00f, 0.00f, //J=
K=	-0.572212468	-0.344915839	-0.572212468f, -0.344915839f, 0.00f, 1.00f, 0.00f, 0.00f, //K=
L=	0.153282769	-0.710083350	0.153282769f, -0.71008335f, 0.00f, 1.00f, 0.00f, 0.00f, //L=
M=	0.047394509	-0.604714471	0.047394509f, -0.604714471f, 0.00f, 1.00f, 0.00f, 0.00f, //M=
N=	0.408119728	-0.967526624	0.408119728f, -0.967526624f, 0.00f, 1.00f, 0.00f, 0.00f, //N=
O=	0.667445795	-0.711771171	0.667445795f, -0.711771171f, 0.00f, 1.00f, 0.00f, 0.00f, //O=
P=	0.920127276	-0.966665767	0.920127276f, -0.966665767f, 0.00f, 1.00f, 0.00f, 0.00f, //P=
Q=	-0.311191200	-0.966878432	-0.3111912f, -0.966878432f, 0.00f, 1.00f, 0.00f, 0.00f, //Q=
R=	-0.209504786	-0.859894314	-0.209504786f, -0.859894314f, 0.00f, 1.00f, 0.00f, 0.00f, //R=
S=	-0.465181101	-0.604049668	-0.465181101f, -0.604049668f, 0.00f, 1.00f, 0.00f, 0.00f, //S=
T=	-0.208359753	-0.348447583	-0.208359753f, -0.348447583f, 0.00f, 1.00f, 0.00f, 0.00f, //T=

4. Insertamos los vértices de acuerdo con los triángulos a generar junto con su color a cada uno de ellos. Cabe mencionar que para realizar los cuadrados o figuras de cuatro lados estas serán divididas en dos triángulos.

//Puntos Cabeza

-0.571514379f, 0.943139226f, 0.00f, 0.5f, 0.0f, 0.5f, //C=0
-0.572411733f, 0.585448842f, 0.00f, 0.5f, 0.0f, 0.5f, //D=1
-0.928394144f, 0.586944576f, 0.00f, 0.5f, 0.0f, 0.5f, //E=2

//Puntos pecho

-0.570733978f, 0.73884424f, 0.00f, 1.0f, 0.5f, 0.0f, //F=3
-0.569932484f, 0.01383689f, 0.00f, 1.0f, 0.5f, 0.0f, //G=4
0.153827133f, 0.014666212f, 0.00f, 1.0f, 0.5f, 0.0f, //H=5

//Puntos Piernas

0.153282769f, -0.71008335f, 0.00f, 0.0f, 0.5f, 1.0f, //L=
-0.569932484f, 0.01383689f, 0.00f, 0.0f, 0.5f, 1.0f, //G=
0.153827133f, 0.014666212f, 0.00f, 0.0f, 0.5f, 1.0f, //H=5

//Puntos Manos

-0.569593663f, 0.162211236f, 0.00f, 1.0f, 0.0f, 0.0f, //I=

-0.824822469f, -0.091923715f, 0.00f, 1.0f, 0.0f, 0.0f, //J=
-0.572212468f, -0.344915839f, 0.00f, 1.0f, 0.0f, 0.0f, //K=

//Puntos Pies

0.047394509f, -0.604714471f, 0.00f, 1.0f, 0.5f, 0.7f, //M=
-0.3111912f, -0.966878432f, 0.00f, 1.0f, 0.5f, 0.7f, //Q=
0.408119728f, -0.967526624f, 0.00f, 1.0f, 0.5f, 0.7f, //N=

//Puntos Bolsa 1

-0.465181101f, -0.604049668f, 0.00f, 0.0f, 1.0f, 0.0f, //S=
-0.208359753f, -0.348447583f, 0.00f, 0.0f, 1.0f, 0.0f, //T=
0.047394509f, -0.604714471f, 0.00f, 0.0f, 1.0f, 0.0f, //M=

//Puntos Bolsa 2

0.047394509f, -0.604714471f, 0.00f, 0.0f, 1.0f, 0.0f, //M=
-0.465181101f, -0.604049668f, 0.00f, 0.0f, 1.0f, 0.0f, //S=
-0.209504786f, -0.859894314f, 0.00f, 0.0f, 1.0f, 0.0f, //R=

////Puntos Cola 1

0.667445795f, -0.711771171f, 0.00f, 1.0f, 1.0f, 0.0f, //O=
0.920127276f, -0.966665767f, 0.00f, 1.0f, 1.0f, 0.0f, //P=
0.153282769f, -0.71008335f, 0.00f, 1.0f, 1.0f, 0.0f, //L=

//Puntos Cola 2

0.153282769f, -0.71008335f, 0.00f, 1.0f, 1.0f, 0.0f, //L=
0.408119728f, -0.967526624f, 0.00f, 1.0f, 1.0f, 0.0f, //N=
0.920127276f, -0.966665767f, 0.00f, 1.0f, 1.0f, 0.0f, //P=

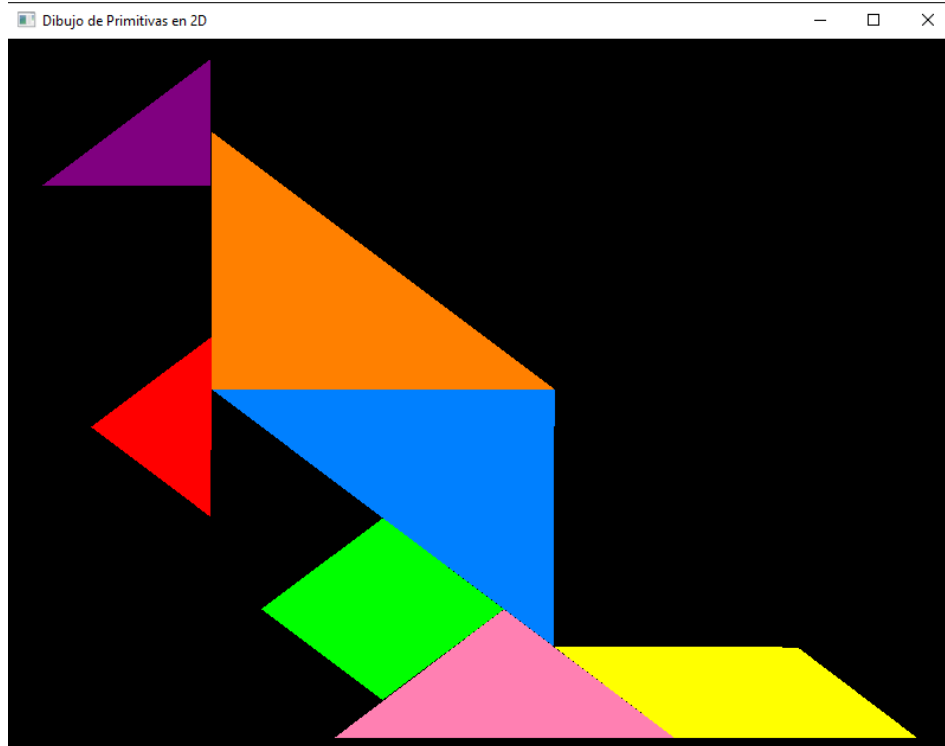
5. Indicamos el orden en el que estarán los vértices.

```
107  unsigned int indices[] = { // note that we start from 0!  
108      0,1,2, //Triangulo Cabeza  
109      3,4,5, //Triangulo Pecho  
110      6,7,8, //Triangulo Piernas  
111      9,10,11, //Triangulo manos  
112      12,13,14, //Triangulo pies  
113      15,16,17, //Triangulo bolsa 1  
114      18,19,20, //Triangulo bolsa 2  
115      21,22,23, //Triangulo cola 1  
116      24,25,26, //Triangulo cola 2  
117  };
```

6. Modificamos para que se vean un total de 27 vértices.

```
170 //glPointSize(5);  
171 //glDrawArrays(GL_POINTS,0,3);  
172  
173 //glDrawArrays(GL_LINES,0,4);  
174 //glDrawArrays(GL_LINE_LOOP,0,4);  
175  
176 //glDrawArrays(GL_TRIANGLES,1,4);  
177 glDrawElements(GL_TRIANGLES, 27, GL_UNSIGNED_INT, 0);  
178
```

7. Ejecutamos.



8. Liga de github.

<https://github.com/refugiorangel/CGIHC/tree/main/Practicas/prac2>

Conclusión.

En esta práctica de laboratorio, generar una imagen mediante figuras geométricas fue relativamente sencillo porque ya existe muchos dibujos en internet con este estilo. Solamente lo que tenemos que hacer es generar la ubicación de los puntos. Para ello, la utilización GeoGebra fue fundamental para tener mayor exactitud de la ubicación de estos. Adicional a esto, Excel también fue una herramienta de mucha ayuda para separar los datos proporcionados por GeoGebra y que sea más fácil insertarlos.

El problema surgió al momento de asignar los colores porque no se tenía el conocimiento para generar estos mediante códigos decimales RGB, pero fue relativamente sencillo encontrar información sobre ello.

Bibliografía.

- Imagen recuperada de <https://mx.pinterest.com/pin/294000681910193223/>
- Colores RGB <https://codigodecolor.com/codigo-de-colores-opengl/>