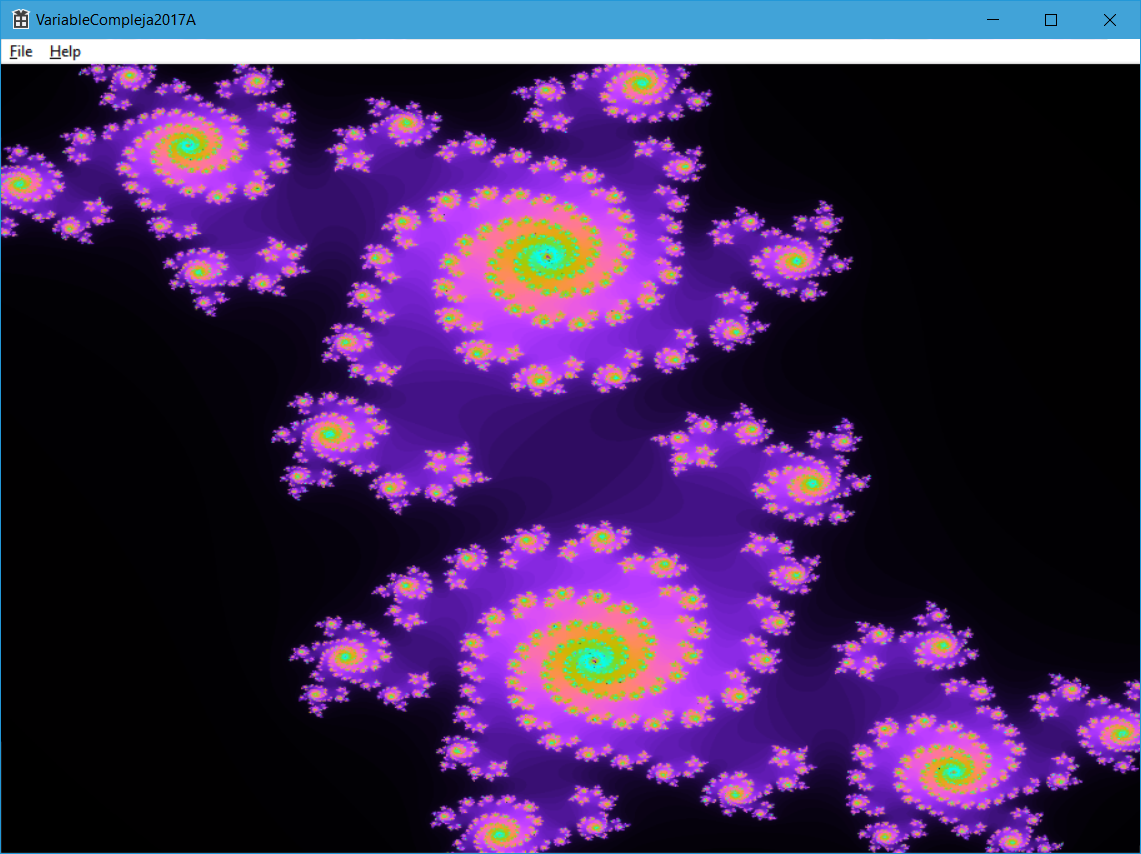
# Tarea 3 Fractales con recurrencias de números complejos: Conjunto de Julia

Construir un shader de computación en HLSL que produzca un fractal de Julia. Un fractal de julia en particular se muestra a continuación:



Esta imagen se produce en tiempos de ejecución mediante la puesta en marcha de un sombreador de computación.

La aplicación permite de interactuar con los valores iniciales de la función auto-recurrente de Julia de tal forma que:

1.- Sea posible construir una gran variedad de conjuntos de julia (diferentes fractales)

2.- Sea posible aumentar o disminuir el tamaño del fractal (zoom-in, zoom-out)

# Entrega:

Subir el proyecto Visual C++ el repositorio de esta tarea.

# Guía de Desarrollo:

## Los números complejos

La cantidad imaginaria fundamental es

i=sqrt(-1)

A partir de la abstracción anterior, un numero complejo se compone de una parte conocida como Real y otra parte conocida como Imaginaria.

Una cantidad C compleja, se puede escribir así:

C= Re + Im\*i Re,Im pertencen a R (Reales)

Cada cantidad compleja está compuesta como una estructura "par ordenado"

de tal manera [Re,Im] se tratase de un vector en R^2.

## SUMA

Operación de suma de dos números complejos:

Sean C1 y C2 dos cantidades complejas [a,b]=a+b\*i y [c,d]=c+d\*i, y realice la suma:

C1+C2=[a,b]+[c,d]=(a+b\*i) + (c+d\*i) = (a+c) + (b+d)\*i = [a+c,b+d]

## MULTIPLICACIÓN

Operación de multiplicación de dos números complejos:

Sean C1 y C2 números complejos a+b\*i y c+d\*i, [a,b],[c,d] respectivamente y realice la multiplicación:

C1\*C2= (a+b\*i) \* (c+d\*i) = a\*c + a\*d\*i + b\*c\*i + b\*d\*i\*i

Observación: Si i=sqrt(-1) entonces i^2=-1

= (a\*c - b\*d)+(a\*d + b\*c)\*i = [a\*c - b\*d, a\*d + b\*c]

## VALOR ABSOLUTO O MODULO

El módulo de un número complejo C denotado por |C| se calcula así:

|C| = sqrt(Re^2(C)+Im^2(C)), es decir, la raíz cuadrada de la suma de cuadrados de las parte real e imaginaria de C.

## FRACTAL DE JULIA

Los fractales basados en números complejos, se describen como funciones complejas autodefinidas o recurrencias (recursiones o con propiedad recursiva). Analizando el comportamiento de la sucesión de números complejos (la serie de números complejos) que producen se pueden definir restricciones que nos ayuden a tomar decisiones. Una de las restricciones interesantes para trazar fractales la prueba de la propiedad de acotamiento.

Una serie de valores se considera “acotada” si para cada elemento en esa serie cumple estar dentro de un rango de valores definidos. Es decir, por ejemplo, la serie de valores de la función seno están acotadas entre -1 y 1, pero no está actotada entre -0.5 y 0. Si una serie no cumple las restricciones de acotamiento, se dice que la serie “diverge” o sale del rango de acotamiento.

En particular la recurrencia conocida como conjunto de Julia:

Zk+1=(Zk)2 + J

Donde Z0=[x,y] cualquier punto del plano complejo comprendido entre -1 y 1 tanto en el eje real como en el imaginario y J cualquier número complejo [r,s] constante entre -1 y 1 tanto en el eje real como en el eje imaginario.

Un punto [x,y] se dice que pertenece al conjunto de Julia (restricción a analizar) si se produce una serie de números complejos comenzando desde Z0=[x,y] hasta Zn y todos estos Zk números cumplen |Zk| < 2. En caso contrario no pertenecen al conjunto bajo las restricciones de magnitud y número N de elementos de la serie.

El pseudo código para trazar el conjunto de Julia en blanco y negro es el siguiente:

1.1) Z=[x,y], i=0, J=[r,s] //[x,y] un punto el plano complejo, [r,s] parámetro constante (offset)

1.2) While(|Z|<2 && i<N)

1.2.1) Z=Z^2+J, i++

1.3) if(|Z|<2)

Pixel[i,j]=Black

else

Pixel[i,j]=White

* Construya un sombreador de computación para que trace el conjunto de julia, decidiendo el color cualquier punto de la superficie trazable en el rango -1 a 1 (tendrá que normalizar el sistema coordenado de pixeles para mapearlo a un sistema de coordenadas dentro de un cuadro de 2 x 2 con centro en 0,0)
* Comience con la versión en blanco y negro, construya su mejora en escala de grises o colores considerando el número de iteraciones i que fueron necesarias para verificar la divergencia o acotamiento.
* Recuerde que los parámetros a controlar por parte de la aplicación serán, la constante inicial J, el zoom y el número máximo de iteraciones N. Al controlar el valor inicial J, se producen los diversos conjuntos de Julia.
* Disfrute el espectáculo visual.

Nota: El número complejo Z0 en el rango normalizado de -1 a 1 en cada componente está en función de la posición del pixel de salida. J es constante controlada por el usuario, también de preferencia en el rango de -1 a 1 en cada componente.