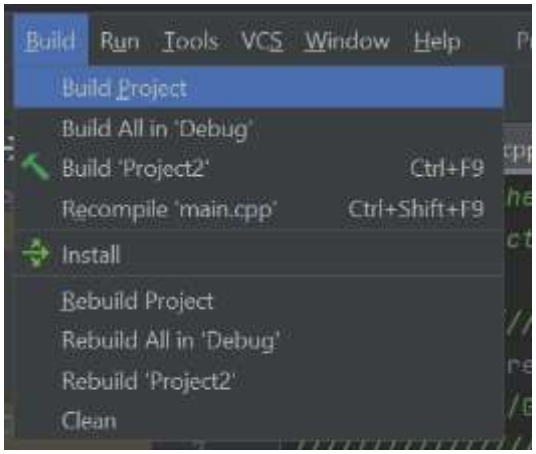
Instructions for Compiling

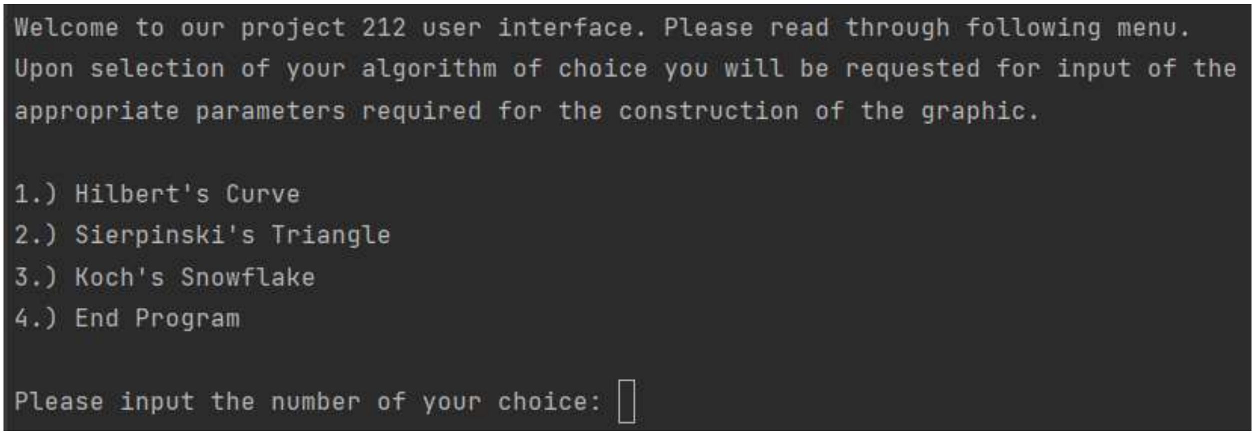
Preliminary Compilation Instructions

The program should be built prior to execution. No commands are necessary to compile in order to run the program.



Running the Program

Upon execution, the main menu will display, prompting the user to select an algorithm for which they wish to generate a graphic. The menu options include generating a Hilbert's Curve graphic (1), a Sierpinski's Triangle graphic (2), a Koch's Snowflake (3), or End Program (4).

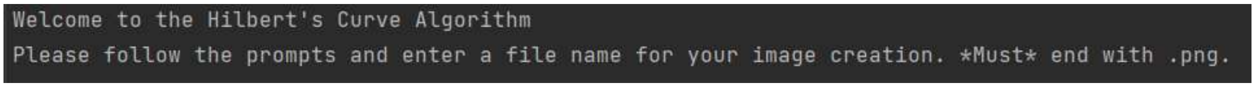


```
Welcome to our project 212 user interface. Please read through following menu.
Upon selection of your algorithm of choice you will be requested for input of the
appropriate parameters required for the construction of the graphic.

1.) Hilbert's Curve
2.) Sierpinski's Triangle
3.) Koch's Snowflake
4.) End Program

Please input the number of your choice: 
```

Once a category to generate a graphic has been selected, a submenu will display prompting the user to enter the specific recursive graphic algorithm's parameters.

Generating a Hilbert's Curve Graphic

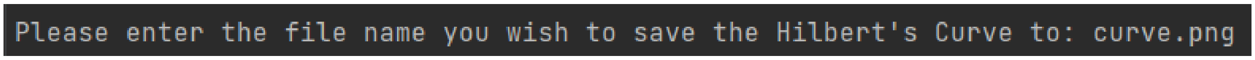
The menu to generate a Hilbert's Curve graphic will display when the user has entered option 1 from the men menu.



```
Welcome to the Hilbert's Curve Algorithm
Please follow the prompts and enter a file name for your image creation. *Must* end with .png.
```

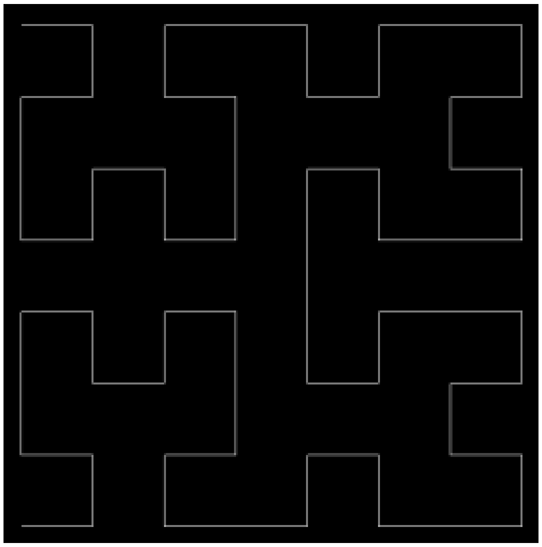
The Hilbert’s Curve submenu will prompt the user to enter a file name.

Note: The file name must contain a .png extension.



```
Please enter the file name you wish to save the Hilbert's Curve to: curve.png
```

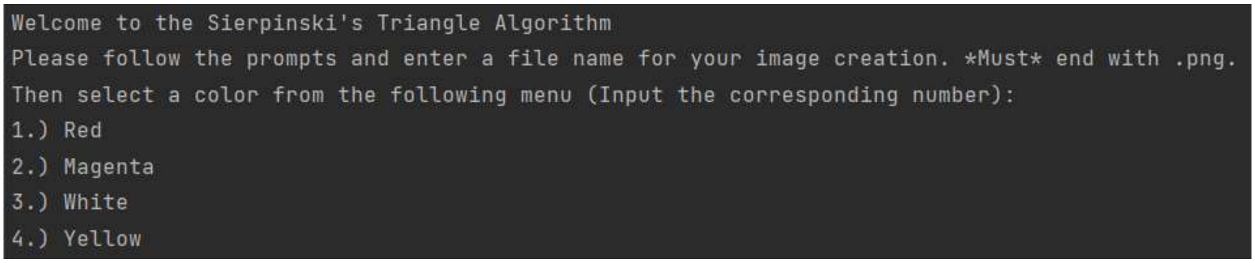
Once the file name has been entered, a window will open displaying the generated graphic and a .png file will be created holding a copy of the graphic image. The illustration above demonstrates a user generating a Hilbert’s Curve graphic which will save to a file named ‘curve.png’. The image below illustrates the graphic generated:



Closing the graphic window will trigger the application to redisplay the main menu where the user may generate another graphic or end the program.

Generating a Sierpinski’s Triangle Graphic

The menu to generate a Sierpinski’s Triangle graphic will display when the user has entered option 2 from the main menu.



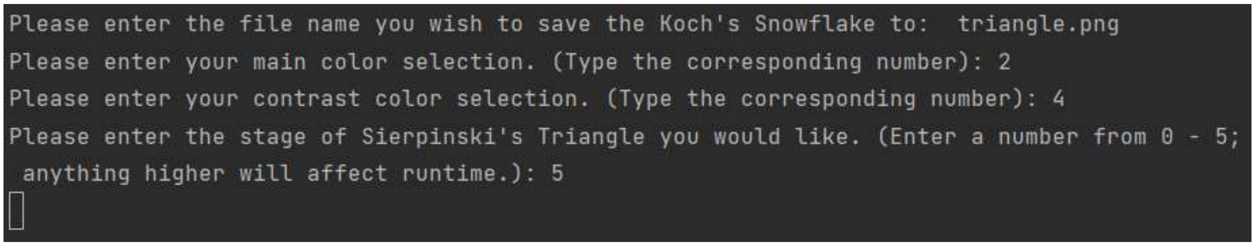
```
Welcome to the Sierpinski's Triangle Algorithm
Please follow the prompts and enter a file name for your image creation. *Must* end with .png.
Then select a color from the following menu (Input the corresponding number):
1.) Red
2.) Magenta
3.) White
4.) Yellow
```

The Sierpinki’s Triangle submenu will prompt the user to enter a file name, select both a main and contrast color from the color menu, and a number to control the number of iterations used to generate the graphic (stage).

Note: The file name must contain a .png extension.

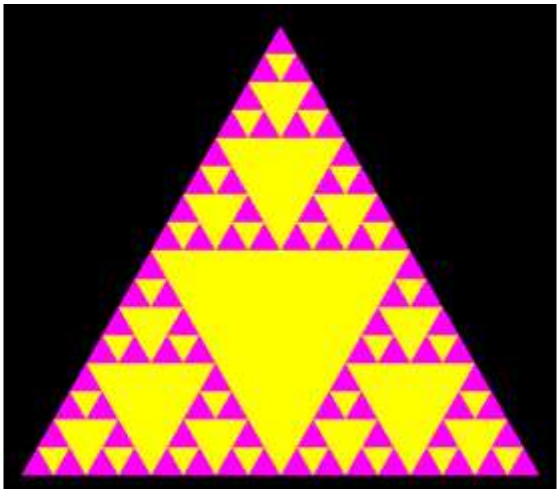
The color options available to generate the triangle graphic include Red (1), Magenta (2), White (3), and Yellow (4).

Note: The stage number must be a number between 0 and 5 (anything higher will distort the resolution and may affect runtime).



```
Please enter the file name you wish to save the Koch's Snowflake to:  triangle.png
Please enter your main color selection. (Type the corresponding number): 2
Please enter your contrast color selection. (Type the corresponding number): 4
Please enter the stage of Sierpinski's Triangle you would like. (Enter a number from 0 - 5;
 anything higher will affect runtime.): 5
```

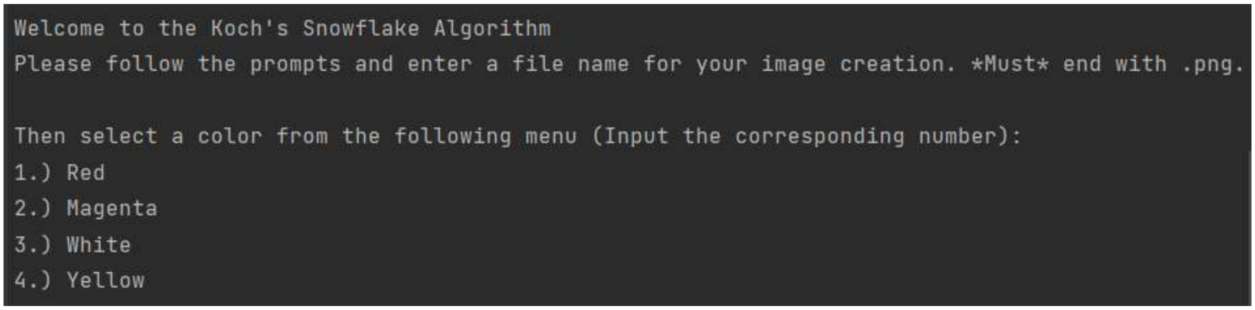
Once these parameters have been entered, a window will open displaying the generated graphic and a .png file will be created holding a copy of the graphic image. The illustration above demonstrates a user generating a Sierpinski's Triangle graphic which will save to a file named 'triangle.png', have a main color of magenta, a contrast color of yellow and terminate its generation at stage 5. The image below illustrates the graphic generated using these parameters:



Closing the graphic window will trigger the application to redisplay the main menu where the user may generate another graphic or end the program.

Generating a Koch's Snowflake Graphic

The menu to generate a Koch's Snowflake graphic will display when the user has entered option 3 from the main menu.



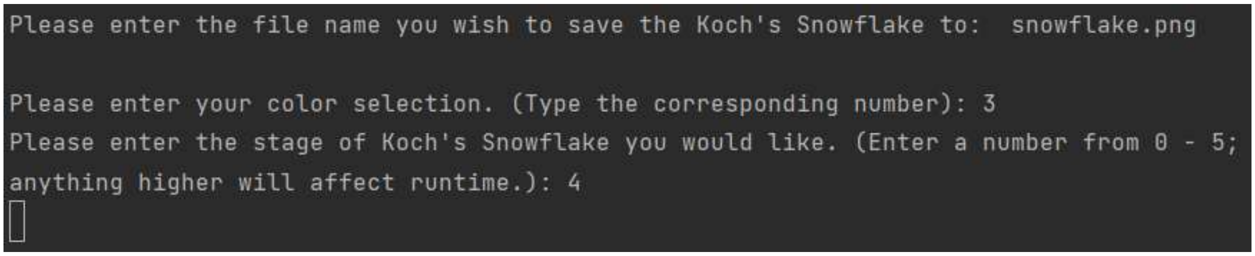
```
Welcome to the Koch's Snowflake Algorithm
Please follow the prompts and enter a file name for your image creation. *Must* end with .png.

Then select a color from the following menu (Input the corresponding number):
1.) Red
2.) Magenta
3.) White
4.) Yellow
```

The Koch’s Snowflake submenu will prompt the user to enter a file name, select a color from a menu of color options, and a number to control the number of iterations used to generate the graphic (stage).

Note: The file name must contain a .png extension.

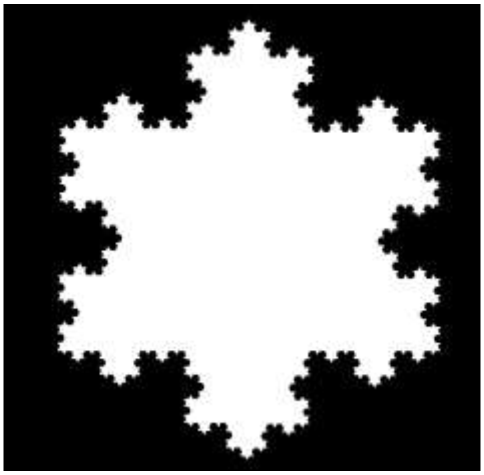
The color options available to generate the triangle graphic include Red (1), Magenta (2), White (3), and Yellow (4).

Note: The stage number must be a number between 0 and 5 (anything higher will distort the resolution and may affect runtime).



```
Please enter the file name you wish to save the Koch's Snowflake to:  snowflake.png

Please enter your color selection. (Type the corresponding number): 3
Please enter the stage of Koch's Snowflake you would like. (Enter a number from 0 - 5;
anything higher will affect runtime.): 4
```

Once these parameters have been entered, a window will open displaying the generated graphic and a .png file will be created holding a copy of the graphic image. The illustration above demonstrates a user generating a Koch’s Snowflake graphic which will save to a file named ‘snowflake.png’, have a color of white and terminate its generation at stage 4. The image below illustrates the graphic generated using these parameters:



Closing the graphic window will trigger the application to redisplay the main menu where the user may generate another graphic or end the program.