Python PolyAngle Tkinter - Design a GUI

Contents

Time required: 90 minutes

Tutorial 1: GUI Design Process with PolyAngle

This tutorial idea comes from dead reckoning planning for a student robot project. We are going to create a Python console program that allows you to enter the number of sides in a regular polygon, then calculate the interior and exterior angles.

The Math

Start with solving the problem, create the algorithm. An algorithm is process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Problem statement: Given the number of sides of a regular polygon, how do we calculate the interior and exterior angles? Let's do the math.

The sum of interior angles in a triangle is 180°. To find the sum of interior angles of a polygon, multiply the number of triangles in the polygon by 180°. The formula for calculating the sum of interior angles is (n - 2) x 180° where n is the number of sides. All the interior angles in a regular polygon are equal.

The formula for calculating the size of an interior angle is:

interior angle of a polygon = sum of interior angles ÷ number of sides.

The sum of exterior angles of a polygon is 360°.

The formula for calculating the size of an exterior angle is:

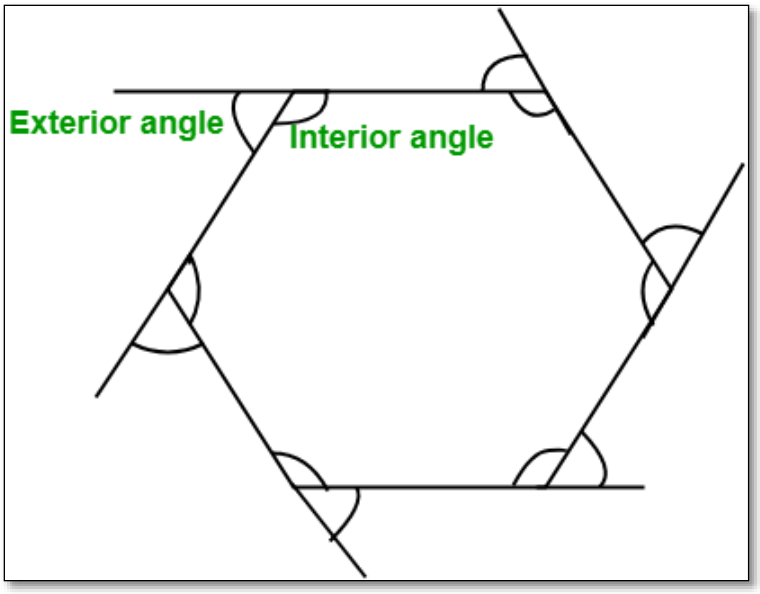
exterior angle of a polygon = 360 ÷ number of sides.

Interior Angle

The angle between two adjacent sides inside the polygon is known as the Interior angle.
Interior Angle = (n-2) x 180 / n

Exterior Angle

The angle formed by any side of a polygon and the extension of its adjacent side is known as the Exterior angle.
Exterior angle =360 / n



Pseudocode

The first step is a high-level look at the program. Think through what you want your program to do as if you were the user running your completed program.

```
Get the number of sides of a regular polygon from the user
Calculate the interior angle
Calculate the exterior angle
Display the results
```

Assignment 1: PolyAngle with tkinter and ttk

Let's convert our polyangle console program into a tkinter GUI program. The Fahrenheit Tutorial conversion gives you a model for this process.

GUI Design Process

Let's go through a process to design and create a simple GUI tool to convert a distance in feet to the equivalent distance in meters.

The high-level pseudocode is the same for a console program as for a GUI program. With a GUI, we also have an interface to design.

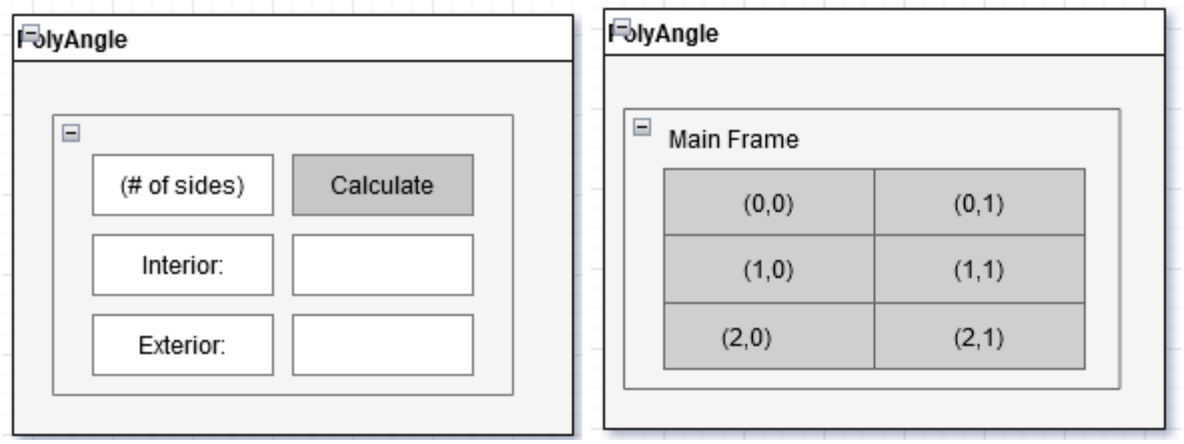
Pseudocode

The first step if the high-level look at the program. Think through what you want your program to do if you were the user running your completed program.

```
Get the number of sides of a regular polygon from the user
Calculate the interior angle
Calculate the exterior angle
Display the results
```

Draw It Out

This is a very rough conceptual sketch of a possible design. Your sketch does not have to look the same.



The Tkinter grid layout manager works very well for most program designs. The following sketch shows how the design might look in a grid layout inside a frame. This visual guide makes it easy to keep track of which row and column each widget is in.

Here are the widgets we will need.

1. LabelFrame to hold all the widgets
2. Text entry to type in the number of sides
3. Calculate button will get the value out of that entry, perform the calculation, and put the result in the two labels below the entry.
4. Two display labels, one for Interior Angle and one for Exterior Angle.

Use the techniques we learned in the Python GUI tutorial Temperature Converter to create a tkinter program like the example run.

1. Create high level pseudocode
2. Break the pseudocode into smaller programmable steps
3. Create a sketch for your user interface
4. Determine the layout manager
5. The polygon.ico file is attached to this assignment. You will have to lookup how to add an ico to a Tkinter program.

Example Run

The first example is tkinter only. The second example adds ttk.

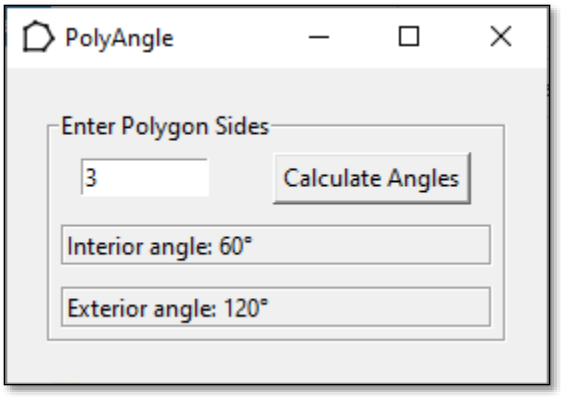


Figure 1 tkinter

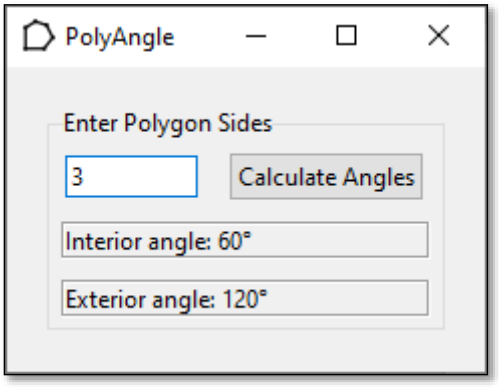
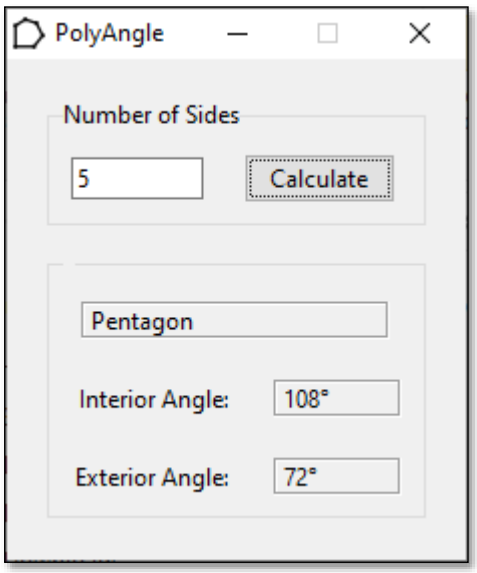


Figure 2 tkinter.ttk

Challenge: This example uses a list of polygon names.



Suggested Activity: Create a Windows Executable with Nuitka

Nuitka will create a standalone Windows executable file from our Python code. Many other methods result in an executable that is detected as a virus.

1. Install the Nuitka package with pip → **pip install nuitka**
2. If haven't already, create a PolyAngle folder for your program files.
3. In your program files folder → create a batch file named **nuitka_gui.bat**
4. Add the following code to the batch file. This nuitka code is specific to tkinter programs.
5. You will want to find an ico file from a web site such as www.iconarchive.com

```
cd c:\temp

python -m nuitka ^
    --mingw64 ^
    --lto=no ^
    --onefile ^
    --disable-ccache ^
    --plugin-enable=tk-inter ^
    --windows-disable-console ^
    --windows-icon-from-ico=polygon.ico ^
    polyangle_gui.py
pause
```

6. Create a **c:\temp** folder to use to create the exe file.
7. Copy the files from your program folder to c:\temp
8. Double click **nuitka_gui.bat**
9. The first time you run nuitka, it will ask you to download and install some compiler tools. Agree to installing the tools.
10. When the process is complete, you will have an exe file. Double click it to run.

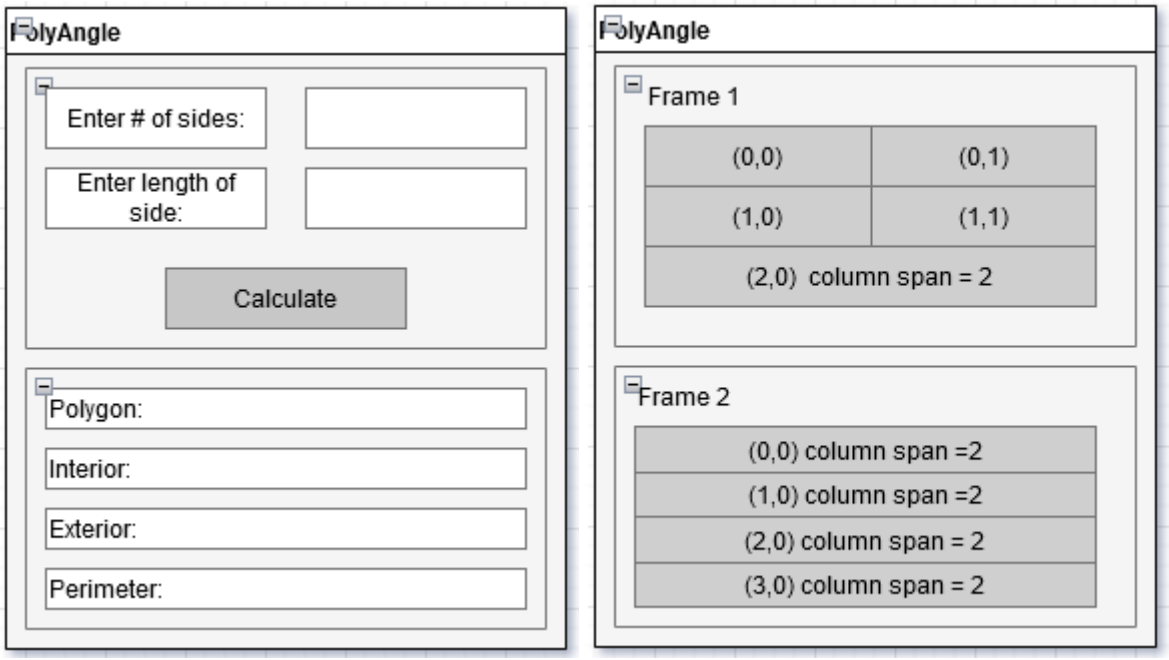
This is the code to build the CLI console program you created earlier into a Windows executable.

```
cd c:\temp

python -m nuitka ^
    --onefile ^
    --mingw64 ^
    --lto=no ^
    --disable-ccache ^
    --windows-icon-from-ico=polygon.ico ^
    polyangle_cli_oop_perimeter.py
pause
```

Assignment 2: Add PolyAngle Perimeter Calculation

Add Polygon perimeter calculation.



Assignment Submission

1. Attach your user interface sketch to the assignment.
2. Attach all python program files to the assignment.
3. Attach screenshots showing the successful operation of the program.
4. Submit in Blackboard