VIDZEME UNIVERSITY OF APPLIED SCIENCES

Vidzeme University of Applied Sciences

Faculty of Engineering

Python OOP and Modelling

group 01

practical work #2

Valmiera, 2021

Table of contents

Document versions			
Version	Status / Changes	Date	Author
1.0	Initial document version	05.12.2021	A.Jenerts

Contacts and responsible (-s)			
Name Surname	Department	Position	Contact information (e-mail)
Signe Grosberga	Group 01	Member	signe.grosberga@va.lv
Edmunds Sprūdžs	Group 01	Member	edmunds.sprudzs@va.lv
Andris Jenerts	Group 01	Member	andris.jenerts@va.lv

1 Use case diagram

2 Drawing n-pointed star with turtle graphics

Task is to make function that draws n-pointed star using Python turtle module. It is stated that n can be any number that is odd. It should be clear that these number can be only integers and no less than 3 because it is impossible to draw star with les then 3 vertices.

2.1 Geometry overview

To better understand process of drawing stars we tried to draw different variants of them on the paper. Drawing 3- and 5-point stars are straight forward but while drawing 7-point star it becomes clear that is it possible to draw such star in multiple ways. Both ways are shown in 1. image below.

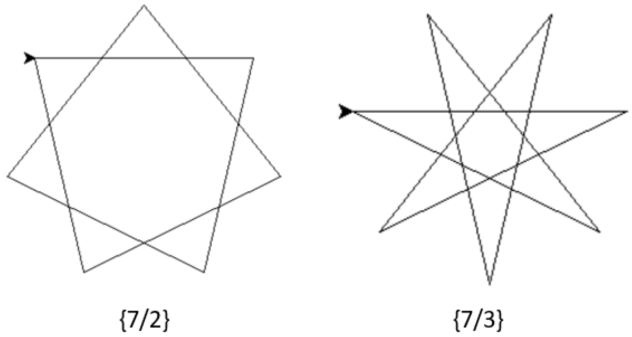

Image 1 Two variants of 7-pointed star

Doing a bit further research team found out that there is special notation for such phenomenon called Schläfli symbols.[1] Not going into deep details this notation consists of two parts:

$$p - \mathit{number\ of\ vertices}$$

$$q - \ \mathit{turning\ number}$$

In general form Schläfli symbol for star looks like this:

$$\{p/q\}$$

To better understand q term one should loop at image below, where turning number is visualized

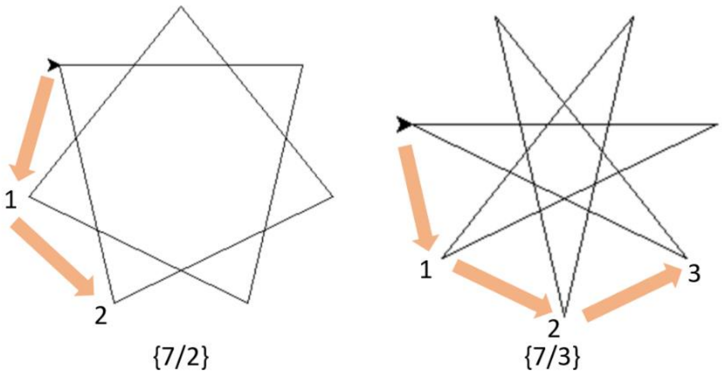

Image 2 Turning number illustration

It is important to look at all possible variants of 7-pointed star to derive general formula for finding how many possibilities there are to draw a star in different ways. In the table below you can see all possible variant noted in Schläfli symbols. To each symbol description is provided that explain is such combination is possible or not.

No.	Schläfli Symbol	Notes
1	$\{7/1\}$	This would result in polygon and polygons are not considered stars
2	$\{7/2\}$	First variant of the star
3	$\{7/3\}$	Second variant of the star
4	$\{7/4\}$	Result in the same variant as $\{7/3\}$
5	$\{7/5\}$	Results in the same variant as $\{7/2\}$
6	$\{7/6\}$	Same as $\{7/1\}$ – not a star
7	$\{7/7\}$	This is a point

It becomes clear that there are n variants for n-pointed stars, but only some of them results in stars, always 2 of them results in polygons and one is a point. Rest of the combinations are valid but are paired meaning that two combinations produce the same variant of the star e.g., $\{7/2\}$ and $\{7/5\}$. From these observations it is possible to create general formula for calculating how many stars are possible to draw:

$$v = \frac{n-3}{2}$$

$$where: n - points\ of\ the\ star$$

NOTE: this formula is intended for odd number n as intended for this task.

There is formula for calculating star angle sum that will be used in our Python function to calculate turning angle. [2]

$$\sum \alpha_0 = (n-2q)180°, where$$

$$\begin{aligned} &\alpha_0 - star\ angle \\ &n - point\ of\ the\ star \\ &q - turning\ number \end{aligned}$$

2.2 Turning angle calculation

To implement angle sum formula, we must look at the way how turtle is moving in our program. Is starts by moving straight and then comes to the vertex of the star. This is the point where it needs to do the turning. This is how turning angle is calculated:

1) Calculate angle of the vertex using star angle sum formula and number of the vertices.

$$\alpha_0 = \frac{(n-2q)180}{n}$$

2) Turning angle β van be calculated using basic geometry shown in image 3.

$$\beta = 180 - \alpha_0$$

$$\beta = 180 - \frac{(n-2q)180}{n}$$

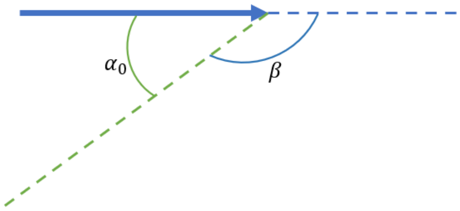

Image 3 Turning angle geometry

Simplifying this equation as shown below it is possible to get simple formula for turning angle β

$$\beta = 180 - \frac{(n-2q)180}{n} \quad \rightarrow \quad \beta n = \frac{180n}{n} - \frac{(n-2q)180}{n}$$

$$\beta n = 180n - 180(n-2q) \quad \rightarrow \quad \beta n = 180(n-n+2q)$$

$$\beta n = 180 * 2q \quad \rightarrow \quad \boldsymbol{\beta} = \frac{\mathbf{360}\boldsymbol{q}}{\boldsymbol{n}}$$

2.3 Star drawing function

Simple function that takes turtle object, number of points and turning number is written to draw a star. This function does not handle any user input errors. Error handling is done outside of the function. As side length is not specified in task it is hard coded and cannot be changed without altering the code.

```
def drawStar(thisTurtle: turtle.Turtle, n, q):
    sideLength = 200
    turnAngle = 360*q/n
    for point in range(n):
        thisTurtle.forward(sideLength)
        thisTurtle.right(turnAngle)
```

2.4 User inputs and error handling

3 Used resources

1. Tychonievich, L. (2012). Schläfli Symbols. Retrieved 5 December 2021, from https://www.cs.virginia.edu/~lat7h/blog/posts/219.html
2. Alsina, C., & Nelsen, R. (2010). *Charming proofs* (pp. 60-61). Washington (DC): M.A.A.
3. Turtle — Turtle graphics — Python 3.10.0 documentation. Retrieved 5 December 2021, from https://docs.python.org/3/library/turtle.html