CS515: Computer Graphics

Semester I, 2020 – 2021

Programming Assignment 1: Turtle Graphics

• Introduction

Turtle Graphics is an alternative positioning system for drawing 2D graphics using a relative cursor ("the turtle") on a Cartesian plane. Imagine a turtle moving around the screen with a pen attached to the bottom of its shell. The turtle's position can be described by a triplet (x, y, θ), giving the location of the center and the orientation of the turtle. The pen itself has attributes such as color, width, etc.

The turtle moves with commands that are ***relative*** to its own current position, such as "move forward 10 units" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it, setting its color, or setting its width.

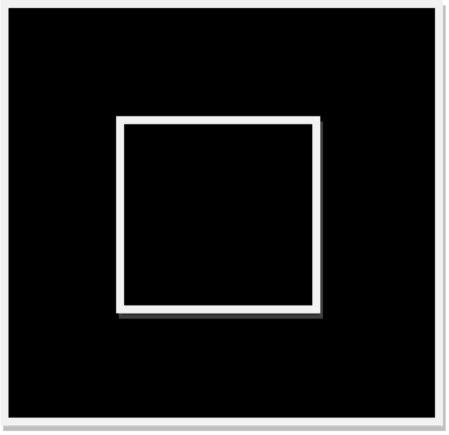
A typical API for such a system includes functions such as:

1. **Init(x,y,theta)** – Initialize or set turtle's position and orientation
2. **Forward(distance)** – Move forward *distance* units
3. **Backward(distance)** – Move backwards *distance* units
4. **Right(angle)** – Turn right *angle* degrees
5. **Left(angle)** – Turn left *angle* degrees
6. **PenDown(yes_no)** – Draw line when moving or move without drawing
7. **LineColor(r,g,b)** – Set current drawing color (r, g, b in range 0.0 – 1.0)
8. **BGColor (r,g,b)** – Set background color (r, g, b in range 0.0 – 1.0)
9. **LineWidth(width)** – Set line width for drawing
10. **ResetPosition** – Set turtle's position and orientation to (0,0,0)
11. **ClearScreen** – Clear the screen

• Example

Draw a square centered at (250,250) and side length of 100 in a window size of 500 x 500:

```
Init( 250, 250, 0 );        // Move to the center (facing right)
BGColor( 0, 0, 0 );         // Set Background color black
LineColor( 1 , 1, 1 );      // Set Line color white
PenDown( False );           // Move without drawing
Forward( 50 );              // Move forward (right) 50 units
Left( 90 );                 // Turn left 90 degrees (facing up)
PenDown ( True );           // Start Drawing
Forward( 50 );              // Move forward (up) 50 units
Left( 90 );                 // Turn left 90 degrees (facing left)
Forward( 100 );             // so on…
```



```
Left( 90 );
Forward( 100 );
Left( 90 );
Forward( 100 );
Left( 90 );
Forward( 50 );
```

- ## Your Task

  1. Implement a turtle graphics library using OpenGL. Your library should include all the functions described above.
  2. Produce correct output for the Square example given above (as a check!)
  3. Produce correct output for instructor's example (Unknown!!)
  4. Create **two** of your own, original 2D drawings (Something interesting and cool!!). At least one of the two drawings should have some curves (circles, etc.)
  5. For testing purpose, your program should accept keyboard input and at each button press should produce one of the above drawings.

- ## Submit

  1. All source files and make files as one tar-gzipped archive.
     a. It should compile without any errors and produce the executable for testing. ***Negative marks for any problems/errors.***
     b. When unzipped, it should create a directory with your ID. Example: 2008CS1001–PA1 (NO OTHER FORMAT IS ACCEPTABLE!!! Case sensitive!!!)
     c. ***Negative marks if the TA has to manually change this to run his/her scripts!!***
     d. Include a README file to convey any details (example, how to compile and run your code, which button press draws what, etc.)
  2. Submit/Upload it to Google Classroom.

- ## Marking

1. Correct implementation of the functions (5 points each) : 55 Marks
2. Correct output for the square example: 5 Marks
3. Original Drawing 1: 15 Marks
4. Original Drawing 2: 15 Marks
5. Correct output for instructor's example: 10 Marks
6. Total = 100
7. **STRICTLY no copying/sharing of source code OR Discussions!.**