******The British College**

**KATHMANDU**

**Coursework Submission Coversheet**(individual coursework only)

**Faculty of Arts, Environment and Technology LBU Student Id:**

C7275280

**For checking by the student:**

Please ensure all information is complete and correct and attach this form securely to the front of your work before posting it in a coursework collection box.

Award name: BSc (Hons) Computing

Module code: 36549

Module name: Advanced Software Engineering

Module run: 2021

Coursework title: Graphical Programming Language Application

Due Date: 26th May, 2021

Module leader: (In LBU) Dr. Duncan Mullier

Module Supervisor: (In TBC) Resham Bahadur Pun

**TURNITIN** Checked: YES NO ***(please circle)***

Submission date & time: Date: 26th May, 2021 Time: 11:30

**Total Word Count: Total Number of Pages (including this front sheet):**

**In submitting this form with your assignment, you make the following declaration:**  
I declare, that the coursework submitted is my own work and has not (either in whole or part) been submitted towards the award of any other qualification either at LBU or elsewhere. I have fully attributed/referenced all sources of information used during the completion of my assignment, and I am aware that failure to do so constitutes an assessment offence.

Signed: Anwesh` ` Date: 26th May, 2021

**You are strongly advised to retain a second copy of your work in case of any query about the assignment.**

1372

14

**For completion by the faculty:**

**This mark is provisional and subject to moderation and approval by the relevant examining board**

**Teacher's Feedback**

**Teacher's Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Table of Contents

[Introduction 2](#_Toc72957537)

[Design Patterns: 3](#_Toc72957538)

[Factory Design Pattern 3](#_Toc72957539)

[About the project 4](#_Toc72957540)

[Single line commands: 4](#_Toc72957541)

[1. Position Pen (Moveto ): 4](#_Toc72957542)

[2. Pen draw (drawTo): 5](#_Toc72957543)

[3. Circle: 5](#_Toc72957544)

[4. Rectangle: 7](#_Toc72957545)

[5. Triangle: 8](#_Toc72957546)

[6. Square: 9](#_Toc72957547)

[8. Clear command: 11](#_Toc72957548)

[Unit Test: 11](#_Toc72957549)

[Advantage of unit testing: 11](#_Toc72957550)

[Version Control: 12](#_Toc72957551)

[Bibliography 13](#_Toc72957552)

# Introduction

**Graphical Programming Language Application** is a simple code editor that lets you write commands and use those commands to draw graphics in the drawing canvas. The application has a typing area for both single and multi-line commands, as well as a window for displaying output, as shown in the screenshot below. A single line editor allows you to write one line of code at a time, whereas a multiple line editor allows you to write multiple lines of code at once and run them all at the same time. If the commands are valid, they will be executed immediately; if they are not, errors will be displayed in the error console area.

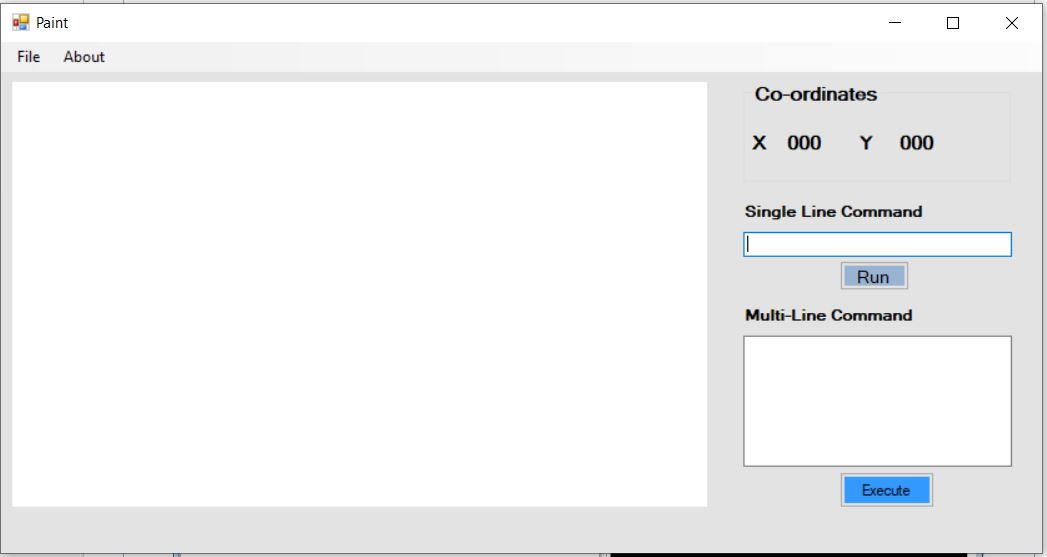


Fig: Interface of Graphical Programming Language Application (GPL)

The application is developed using the concepts of GDI (Graphical Device Interface) in C# .Net language. “GDI is the portion of the Windows operating system that provides two-dimensional vector graphics, imaging, and typography. GDI+ improves on GDI (the Graphics Device Interface included with earlier versions of Windows) by adding new features and by optimizing existing features.”(De George, 2017)

# Design Patterns:

**“Design patterns** are typical solutions to common problems in software design. Each pattern is like a blueprint that you can customize to solve a particular design problem in your code.” (Design Patterns, 2021) In simple terms, a design pattern is a reusable, flexible solution to software design issues that arise frequently during the development of real-world applications. It's a model or description of how to solve problems that can be used in a variety of situations. The complexity, level of detail, and scope of applicability of Design Patterns vary. Design patterns are divided into three categories as a result of this: Patterns of creation, behavior, and structure.

The design pattern that the Graphical Programming Language application follows is **Factory Design Pattern,** which comes under creational pattern that defines an interface for creating an object but allows the classes that implement the interface decide which class to instantiate.

# Factory Design Pattern

According to Gang of Four Definition “Define an interface for creating an object, but let the subclasses determine which elegance to instantiate. The Factory method lets a class defer instantiation it uses to subclasses”.

In simple term, factory design pattern is a creational design pattern that provides an interface for creating objects in superclass but allows derived or subclasses to alter the type of objects that will be created.

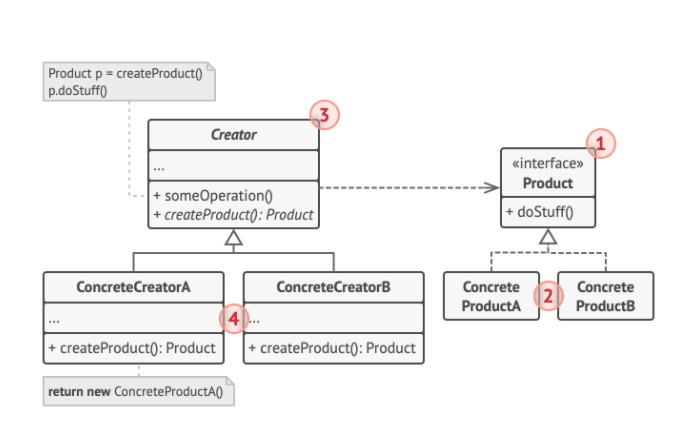


Fig: Structure of Factory Design pattern

The Product class declares the common interface for all objects, whereas concrete products are implementations of the product interface. The concrete creators override the base factory method to return a different type of product, while the creator class declares the factory method that returns new product objects.

# About the project

There are three main executing commands in my project: run, clear, and reset. If the passed commands are valid, the Run command executes the command typed in the typing window; otherwise, it throws an error. The clear command clears the drawing canvas and finally the reset command resets the pen position to its initial axis i.e. x=0 and y =0.

# Single line commands:

The GPL application allows us to write single line commands, which means commands that can be executed one at a time. We can draw different shapes with the color we want as shown below:

## Position Pen (Moveto ):

The moveto command is a single line command in our application that helps to position the pen based on two parameters i.e. x-axis and y-axis.

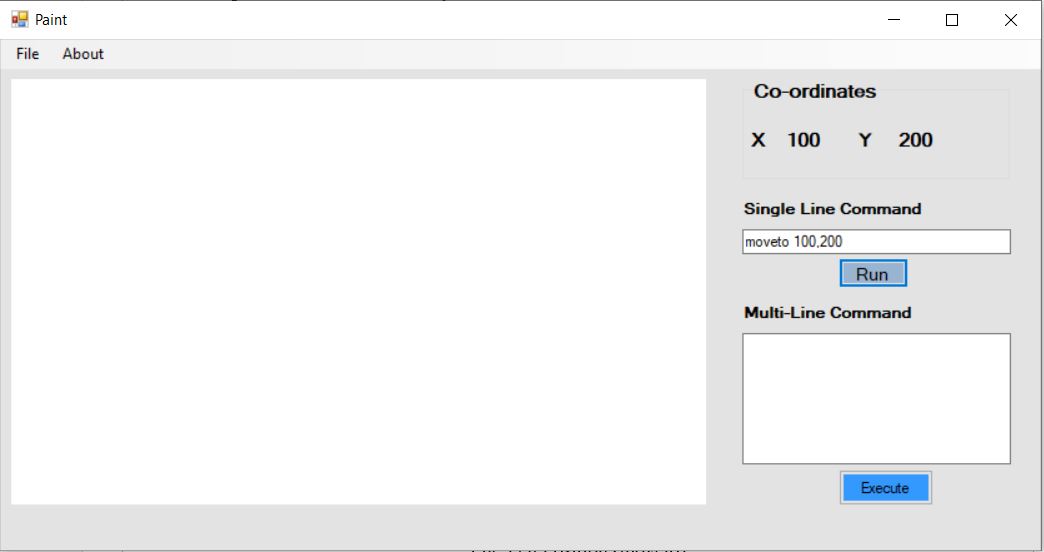


Fig: Pen Position (moveto)

## 2. Pen draw (drawTo):

The drawto command is a single line command in our application that helps to draw line from x to y axis along with the choice of pen color we pass as one of the parameter.

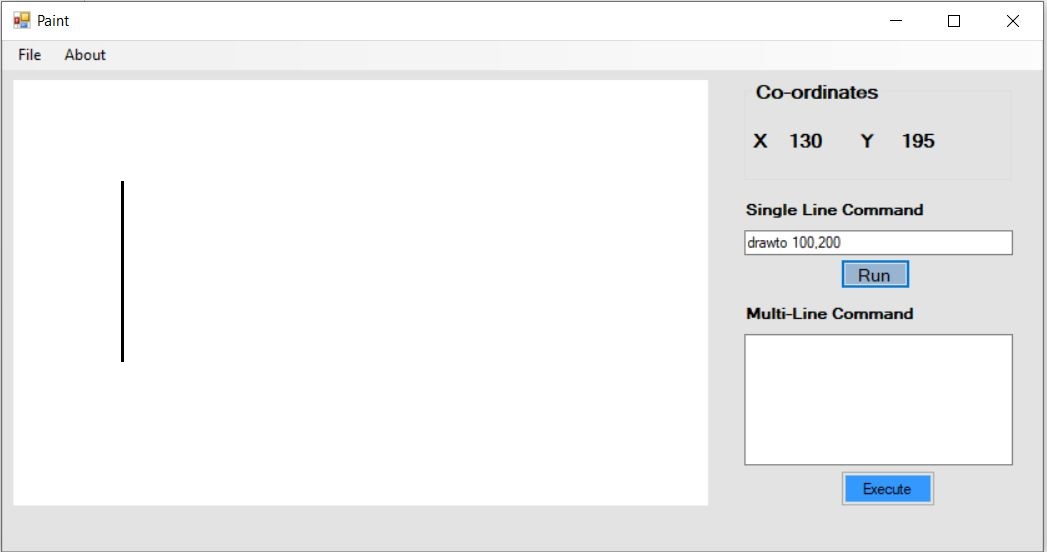


Fig: Pen drawto command

## 3. Circle:

Circle command is a single line command that takes two parameters- color and radius. We can pass color and radius of our choice. If the parameters are missed, the application throws an error.

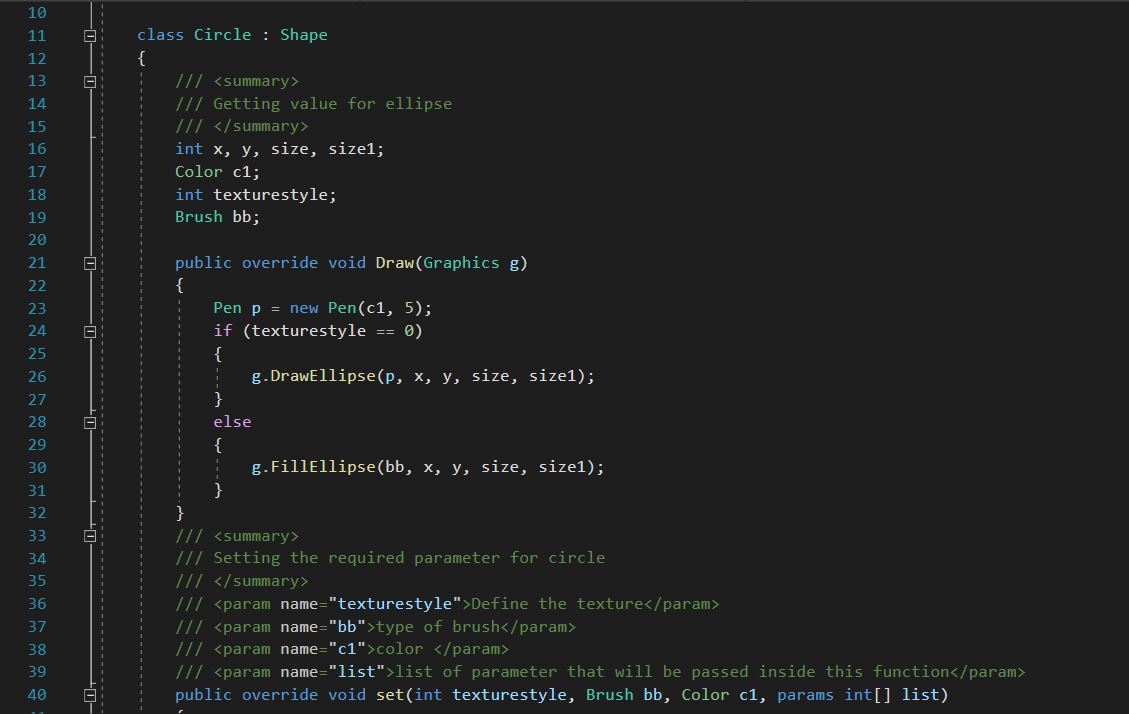
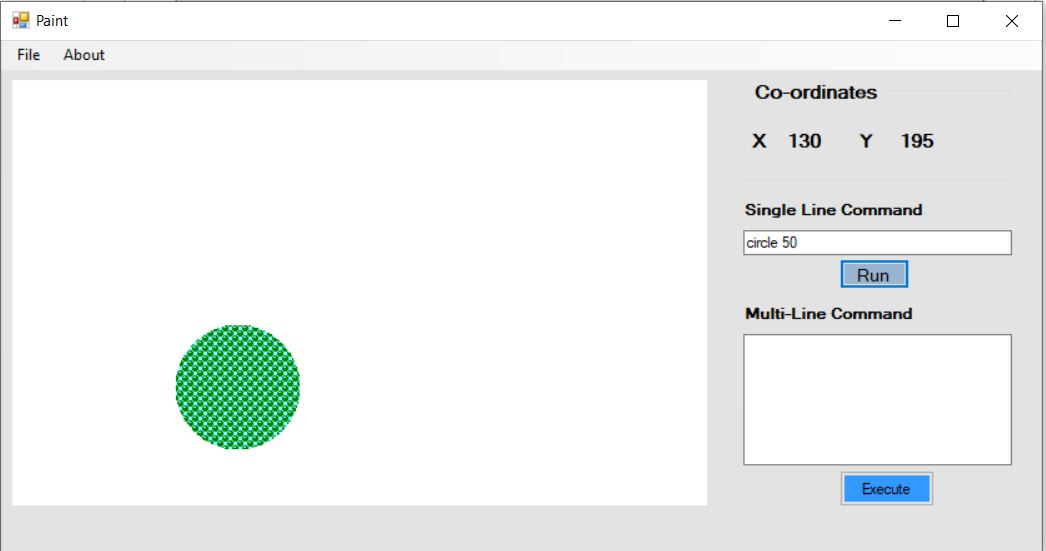


Fig: Snapshot of code for Circle

## 4. Rectangle:

Rectangle command is a single line command that takes three parameters- color, height and width. We can pass color, height and width of our choice. If the parameters are missed, the application throws an error.

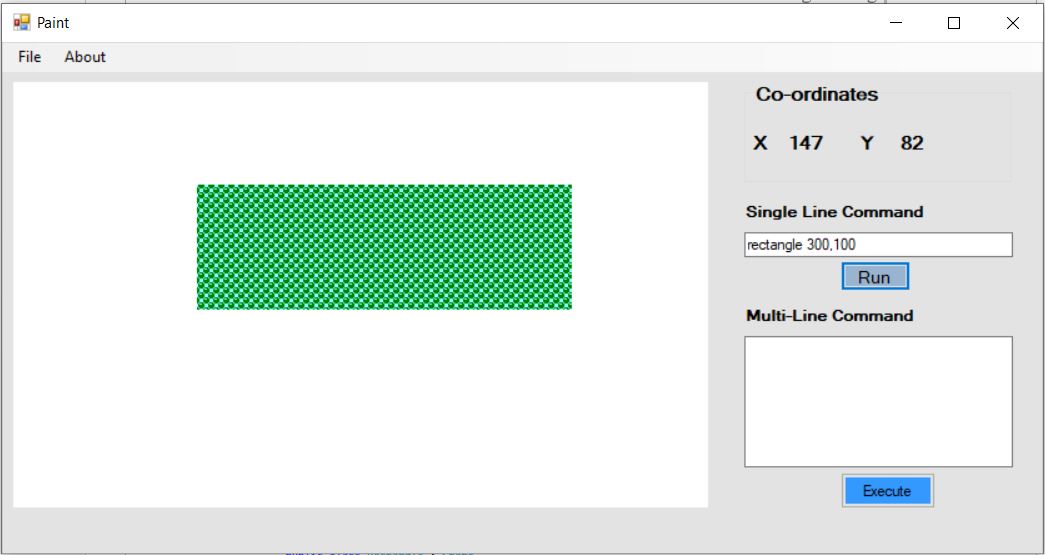


Fig: Snapshot showing output of rectangle command

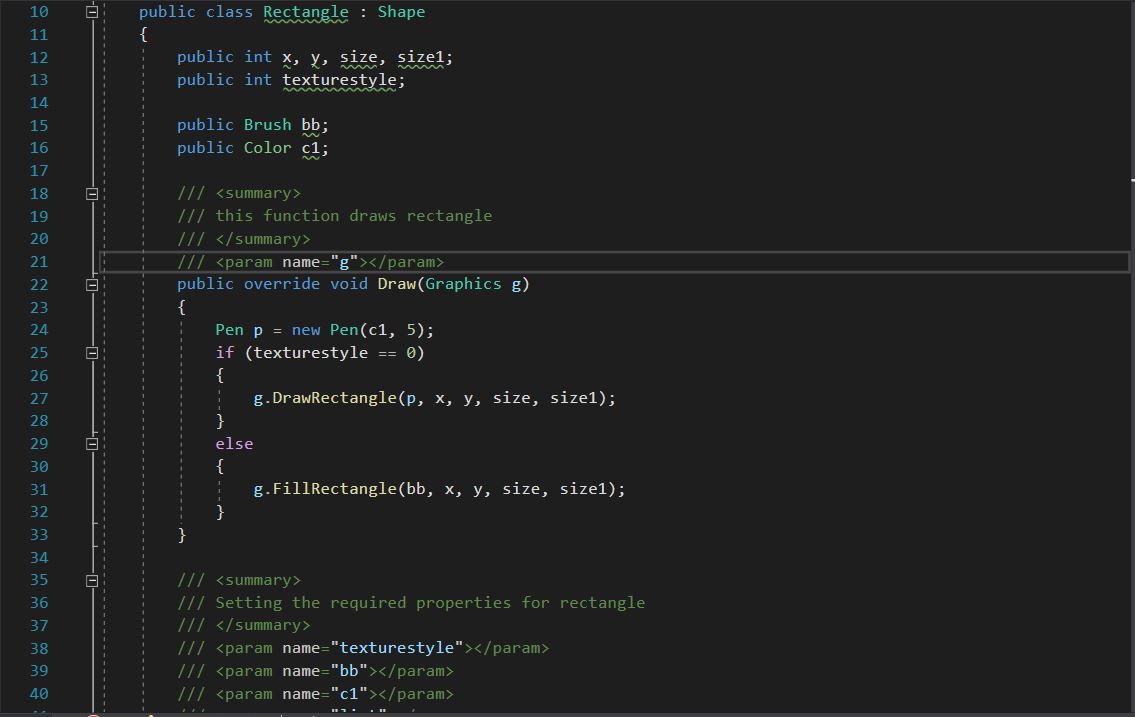


Fig: Snapshot of code for Rectangle

## 5. Triangle:

Triangle command is a single line command that takes four parameters- colors and points. We can pass color, points in order to draw a triangle. If the parameters are missed, the application throws an error.

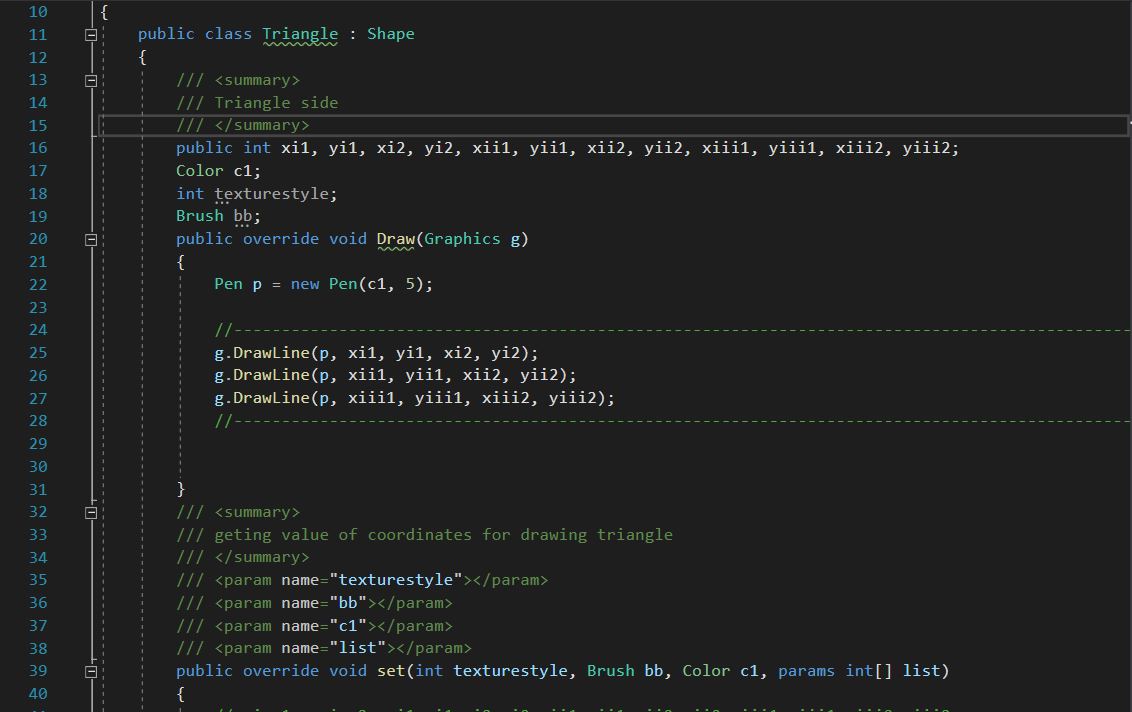


Fig: Snapshot of code for Triangle

## 6. Square:

Square command is a single line command that takes two parameters- color and sides. Since square has equal sides, we can pass color and only one side as a parameter. If the parameters are missed, the application throws an error.

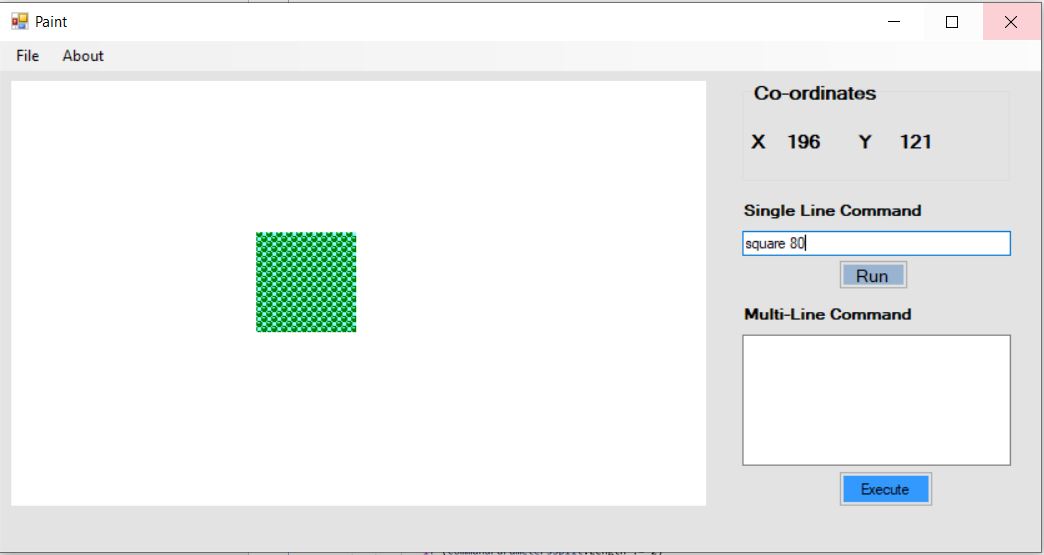
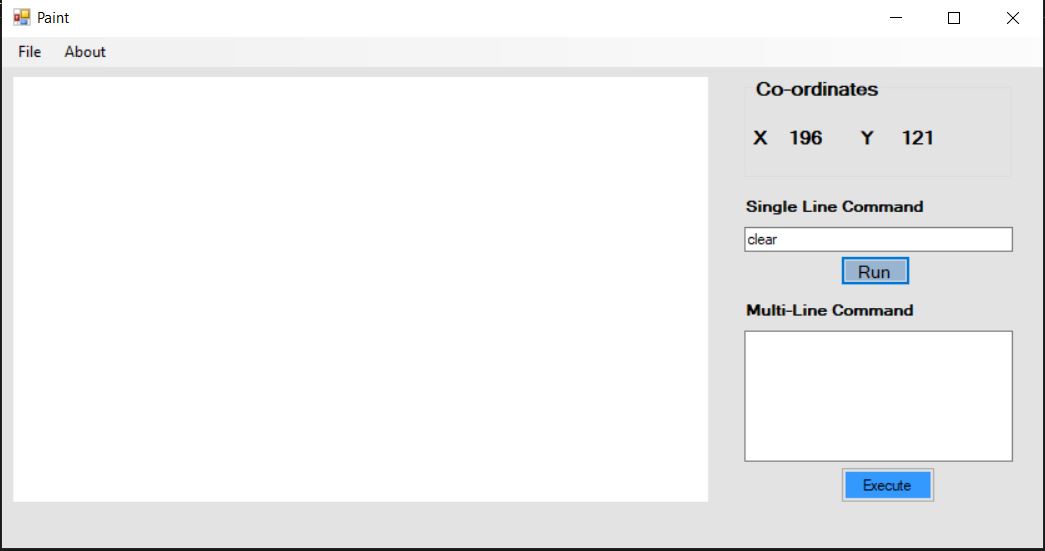


Fig: Snapshot showing snippet of code for square

## 8. Clear command:

The clear commands clears the drawing canvas. When clicked on ok, the canvas is cleared.



# Unit Test:

A unit test is a piece of test code written by any programmer to check the functionality of small parts of larger systems. The goal of unit testing is to keep things simple. In this context, a "UNIT" is the smallest component of a large code section that can be tested, usually a method among many methods of a class. This test ensures that the source code meets the requirements, is well-designed, and performs as expected.

## Advantage of unit testing:

* Reduces defects in the newly developed features or reduces bugs when changing the existing functionality.
* Reduces cost of testing as defects are captured in very early phase.
* Improves design and allows better refactoring of code.
* Unit tests, when integrated with build gives the quality of the build as well.

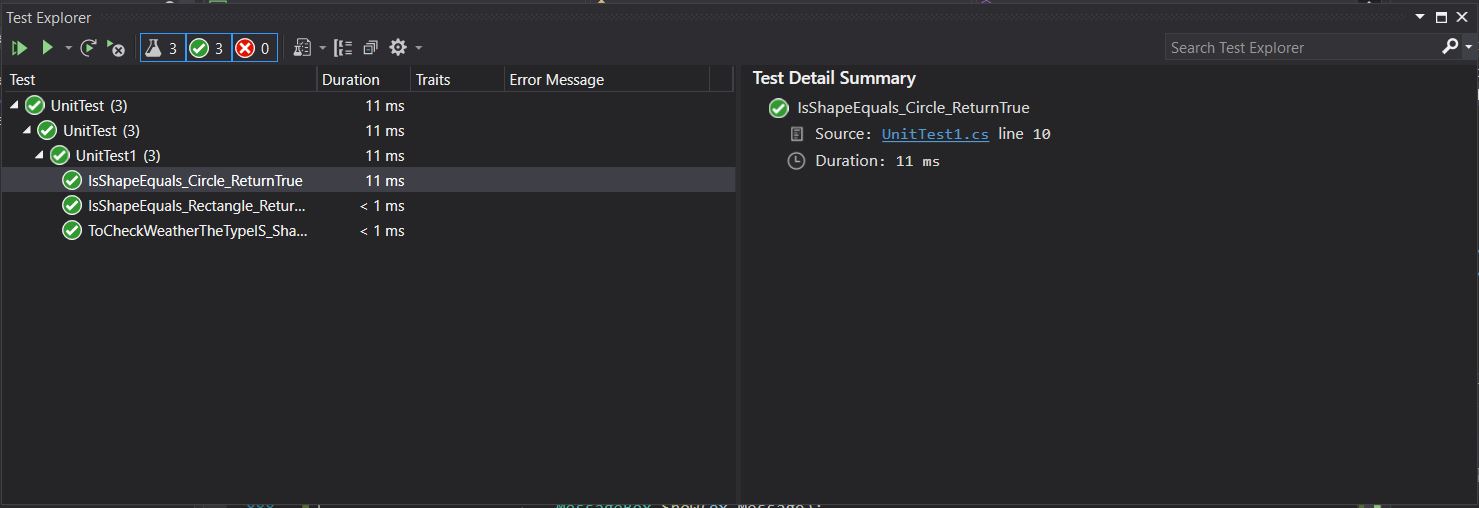


Fig: GPL Unit Test

The above snapshot shows that the unit test performed on the GPL has passed and the code is working efficiently.



Fig: Unit testing with prerequisites.

# Version Control:

The version control of the Graphical Programming Language Application has been done in <https://github.com/>. The following snapshot shows the number of commits made:

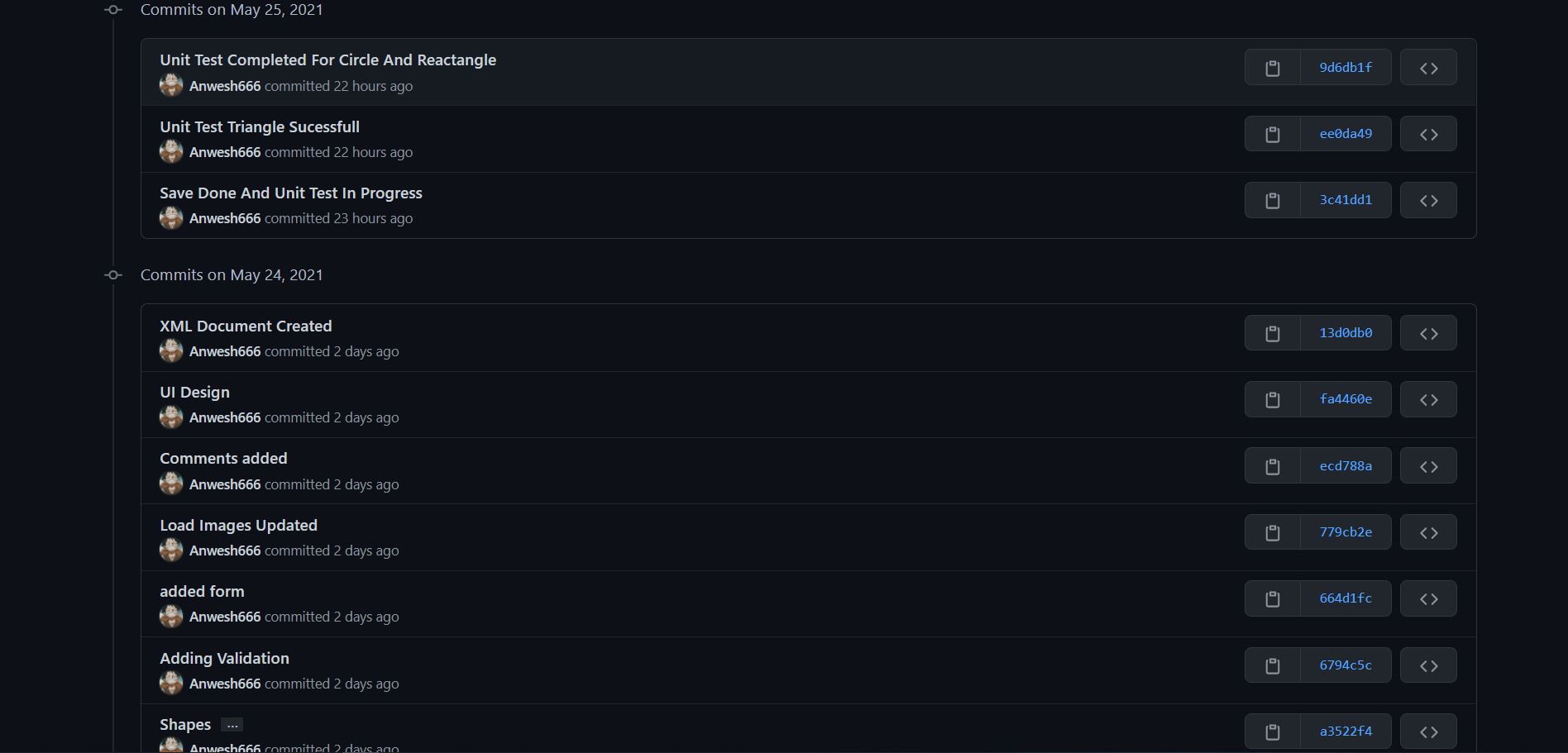


Fig: Snapshot of Total Commits made for Component 1

Github link: https://github.com/Anwesh13/2ASE-comp1and2

# Bibliography

De George, A., 2017. About GDI+ Managed Code - Windows Forms .NET Framework. [online] Docs.microsoft.com. Available at: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/advanced/about-gdi-managed-code?view=netframeworkdesktop-4.8> [Accessed 26 May 2021].

Refactoring.guru. 2021. Design Patterns. [online] Available at: <https://refactoring.guru/design-patterns> [Accessed 26 May 2021].