Sketchpad Language

Zhongyu Wang zw2259

Yichen Liu, yl2904

Yan Peng , yp2321

Motivation

Sketchpad language is designed to construct geometric graphics. The idea comes from Geometer's Sketchpad, a popular commercial interactive geometry software for exploring Euclidean geometry, algebra, calculus, and other areas of mathematics.

Geometer's Sketchpad was created by Nicholas Jackiw, who won Turing Award for the innovation. Geometer's Sketchpad includes the traditional Euclidean tools of classical geometric constructions; that is, if a figure can be constructed with compass and straight-edge, it can also be constructed using this program. [1]The program also allows the determination of the midpoint and mid segments of objects.

However, the software comes in a MS-paint-like GUI where users use toolbars and menus to finish all construction in an interactive way. This is intuitive but due to the limitation of classic geometry constructions, it may be too trivial for users to create a complex figure by only using these menus and buttons.
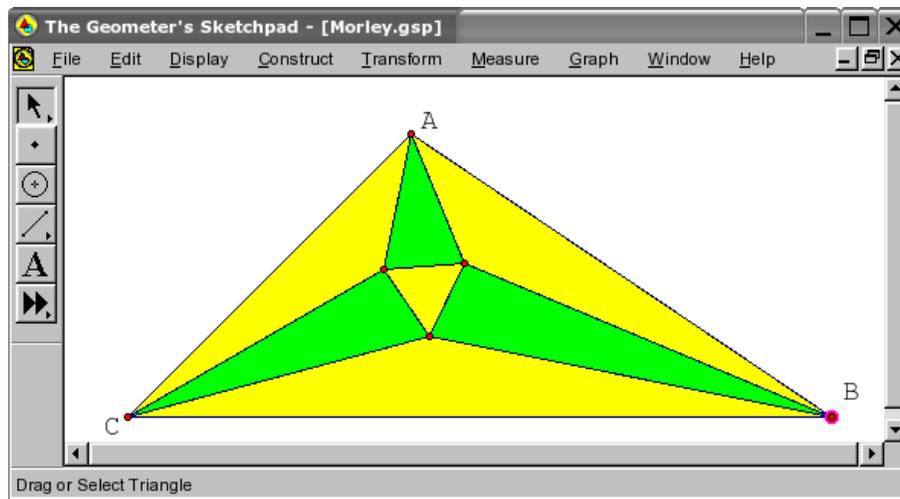
Figure 1 Geometer's Sketchpad [1]

Key Features

Sketchpad language, a formal programming language is designed to construct geometric graphics. The new programming language can complete almost all most important operations in Geometer's Sketchpad. While without using the menus and buttons on Geometer's Sketchpad manually , Sketchpad

language makes it possible for users to use conditional statements like if-then and for and while statements to construct figures.

Also, it is possible to break the limitation of classical geometry by using our non-Euclid functions and objects. It can support both procedure oriented programming and object oriented programming.

Syntax Example

Sketchpad language leverages a very powerful basic type that allows the user to create more complex and interesting components using even more expressive operators.

```
//Below lines draw a center of gravity of a given triangle.

float x1,x2,x3,y1,y2,y3;

point A, B, C, D;

A=getPoint(x1,y1);

B=getPoint(x2,y2);

C=getPoint(x3,y3);

D=Intersect(getLine(A,midpoint(B,C)),getLine(B,midpoint(C,A)));


// Logarithmic spiral, which is not a traditional geometric concept.

int t;

point A,B;

A=getPoint(0,0);

for t=0 to 1000

{

        x=exp(t/100)*cos(t/100);

        y=exp^(t/100)*sin(t/100);

        B=A;

        A=getPoint(x,y);

        getLine(A,B);

};
```

Interest thing our program can realize:

Snowflake Fractal. Our program can draw figure of snowflake fractal with very little work.

Plane Geometry Problems. If a teacher need to prove or disprove a plane geometric proposition, he can use our program to check whether it is true by moving unrestricted points into different position and test if the proposition hold for different situations.

Basic Description

The language have general mathematical and logical operations like most of the popular programming languages.

The core object of Sketchpad Language is basic geometrical elements like points, lines and circles, corresponding to all what compass and straight can do. Very few though, as proved by mathematicians, they are strong enough to finish complex work. A basic library of function are provided to realize basic drawing and measuring function using which users can construct geometrical conception.

The most fascinating property is that if a point is modified, all lines and other elements depend on the point will change in consistency with the modification. The midpoint of a line will always be the midpoint of that line no matter what is done to the figure. This makes our program useful in geometry. In order to realize this property, the relationship of all objects are maintained in a graph structure.

If possible, we'll add object-oriented property to the program, which will greatly increase its ease of use.

Reference

[1] Wikipedia http://en.wikipedia.org/wiki/The_Geometer's_Sketchpad