ALG - A Language for Geometry
Introduction to ALG
Introduction to ALG

What’s ALG

• The attributes of figures

• The relationship between them

• Place action on them
Introduction to ALG

Why ALG?

A  Helpful
B  Simple
C  Fun
Tutorial to ALG
What ALG Can Do?

- **Basic Calculation**
  - Arithmetic Calculation, String Calculation,
  - Boolean Calculation, Basic Control Flow

- **Geometry Calculation**
  - L1//L2(parallel), L1^L2(intersection point)
  - L1|-e1, e1|-e2(positional relation)
  - poly1~poly2(congruent),
    poly1~poly2(similar)
  - Area, Perimeter, Draw
  - Perimeter comparison: << >> <<<-
    >>=
  - Area comparison: ~< ~> ~~< ~~~>
Tutorial to ALG

How To Use

Environment:
gcc, opencv, pkgconfig

Steps:
• make clean
• make
• bash
  ./compiler.sh
  input.alg
  output_alg
def int main()
{
    line 11;
    line 12;
    line 13;
    point p1;
    point p2;
    point p3;
    polygon poly;
    l1=[[10;10],[20;20]];
    l2=[[0;0],[20;1]];
    l3=[[15;20],[20;0]];
    Draw(l1);
    Draw(l2);
    Draw(l3);
    (!display the information of the three lines!)
    display(l1);
    display(l2);
    display(l3);
    p1=l1^l2; !!p1 is the intersect point of 11 and 12
    p2=l2^l3; !!p2 is the intersect point of 12 and 13
    p3=l3^l1;     !!p3 is the intersect point of 11 and 13
    poly=[p1,p2,p3]; !! poly is the intersect area of the three lines
    print("the area of the intersection area is ");
    print(Area(poly)); !!print the area of the intersection part of the three lines
    print_newline();
}
This is a line!
The first point is (10.000000 10.000000)
The second point is (20.000000 20.000000)
This is a line!
The first point is (0.000000 0.000000)
The second point is (20.000000 1.000000)
This is a line!
The first point is (15.000000 20.000000)
The second point is (20.000000 0.000000)
the area of the intersection area is 150.123
def int main()
{
    ellipse e1;
    line l1;
    int i;
    i=50;
    e1=[[0;0],10,10];
    l1=[[50;0],[50;1]];
    Draw(e1);
    while(i>=0)
    {
        Draw(l1);
        if((l1-e1)=="tangent")
        {
            print("tangent");
            print_newline();
            display(l1);
            done;
        }
        Move(l1,-5,0);
    }
Project Architecture
Project Architecture
Project Architecture

Phase 1: Semantic Test
Phase 2: Basic Functions Test
Phase 3: Advanced Functions Test
Summary & Lessons
Summary & Lessons

1. Well design saves time!
2. Get everyone involved!
3. Stick to your goal!
4. Tech guys love tools!
5. Stay synchronized with Stephen!

Yes! We make it!

WHY NOT?

Change the course name as PLT & O’Caml? You take one but get both!
Thank you!