

EasKey

A language for mouse/keyboard operation

Keqiu Hu
Xiaoyu Huang
Jinqi Huang
Zongheng Wang
Lizhong Zhang

Columbia University 2012

Outline

- ① Motivation
- ① Feature
- ① Tutorial
- ① Architecture
- ① Lessons
- ① Conclusion

Motivation

Motivation

1 Auto course registration

REGISTRATION CLASS QUERY

Viewing Options

Hide my name and personal data

Change...

Add or Check Class Status

Call#	Class/Title	Instructor	Points	Pass/Fail	Instructor Permission	Action
93401	COMS 3136 W 001 "ESSENTIAL DATA STRUCTURE"	Lee, Jae W	4.00	Grade ▾	Not Required	Add Class

[Back To Registration](#) [List All Sections](#) [Courseworks](#)

Motivation

1 Auto course registration

2 Get higher score in games like whack a mole



Motivation

- 1 Auto course registration
- 2 Get higher score in games like whack a mole
- 3 Other iterative mouse and keyboard input like entering CD-Key for software setup automation.

Feature

- Abundant Data Types

- string, int, char, float, boolean, color, point, key

- Built-in Windows API Functions

- `__leftClickPoint` `__keyStroke` `__movePoint`

- `__leftDown` `__getPoint` `__getColor` and so on

- User-Friendly Programming Style

Tutorial--gcd function

```
function int gcd(int a, int b)
  while (a != b)
    if (a > b)
      a = a - b;
    else
      b = b - a;
    end if
  end while
  return a;
end function
```

```
function int main()
  int i;
  i = gcd(16,12);
  return 0;
end function
```


Tutorial--gcd function

```
function int gcd(int a, int b)
  while (a != b)
    if (a > b)
      a = a - b;
    else
      b = b - a;
    end if
  end while
  return a;
end function
```

```
function int main()
  int i;
  i = gcd(16,12);
  return 0;
end function
```

```
glb 0
jsr 28
hlt
```

```
ent 0
bra 16
lfp -3
lfp -2
gt
beq 7
lfp -3
lfp -2
sub
sfp -3
pop 1
bra 6
lfp -2
lfp -3
sub
sfp -2
```

```
pop 1
lfp -3
lfp -2
neq
bne -18
lfp -3
rts 2
psi 0
rts 2
ent 1
psi 0
psi 12
psi 16
jsr 3
sfp 1
pop 1
psi 0
rts 0
psi 0
rts 0
```

Tutorial -- More complex examples

```
function int main()
```

```
    int a;
```

```
    point p1 = {12,24};
```

```
    point p2 = {23,46};
```

```
    color c = #203,245,11#;
```

```
    key k = $Space$;
```

```
    point p = p1 + p2;
```

```
    for(int i = 0; i < 10; i = i+1)
```

```
        __print("COMSW4115");
```

```
    end for
```

```
    int i;
```

```
    switch (i)
```

```
        case 1: a = 1;
```

```
        case 1: a = 2;
```

```
        case 1: a = 3;
```

```
        default: a = 4;
```

```
    end switch
```

```
    return 0;
```

```
end function
```

Tutorial -- A program for playing whack a mole

```
function int main()
    while(true)
        __delay(5000);
        color c=#0,54,99#;
        point p1={600,320};
        color c1;
        point p2={800,320};
        color c2;
        point p3={1000,320};
        color c3;
        point p4={600,450};
        color c4;
        point p5={800,450};
        color c5;
        point p6={1000,450};
        color c6;
        point p7={600,600};
        color c7;
        point p8={800,600};
        color c8;
        point p9={1000,600};
        color c9;

        c1=__getcolor(p1);
        c2=__getcolor(p2);
        c3=__getcolor(p3);
        c4=__getcolor(p4);
        c5=__getcolor(p5);
        c6=__getcolor(p6);
        c7=__getcolor(p7);
        c8=__getcolor(p8);
        c9=__getcolor(p9);

        if(c1!=c)
            __movePoint(p1);
            __leftClickPoint(p1);
            c1=c;
        end if
        if(c2!=c)
            __movePoint(p2);
            __leftClickPoint(p2);
            c2=c;
        end if

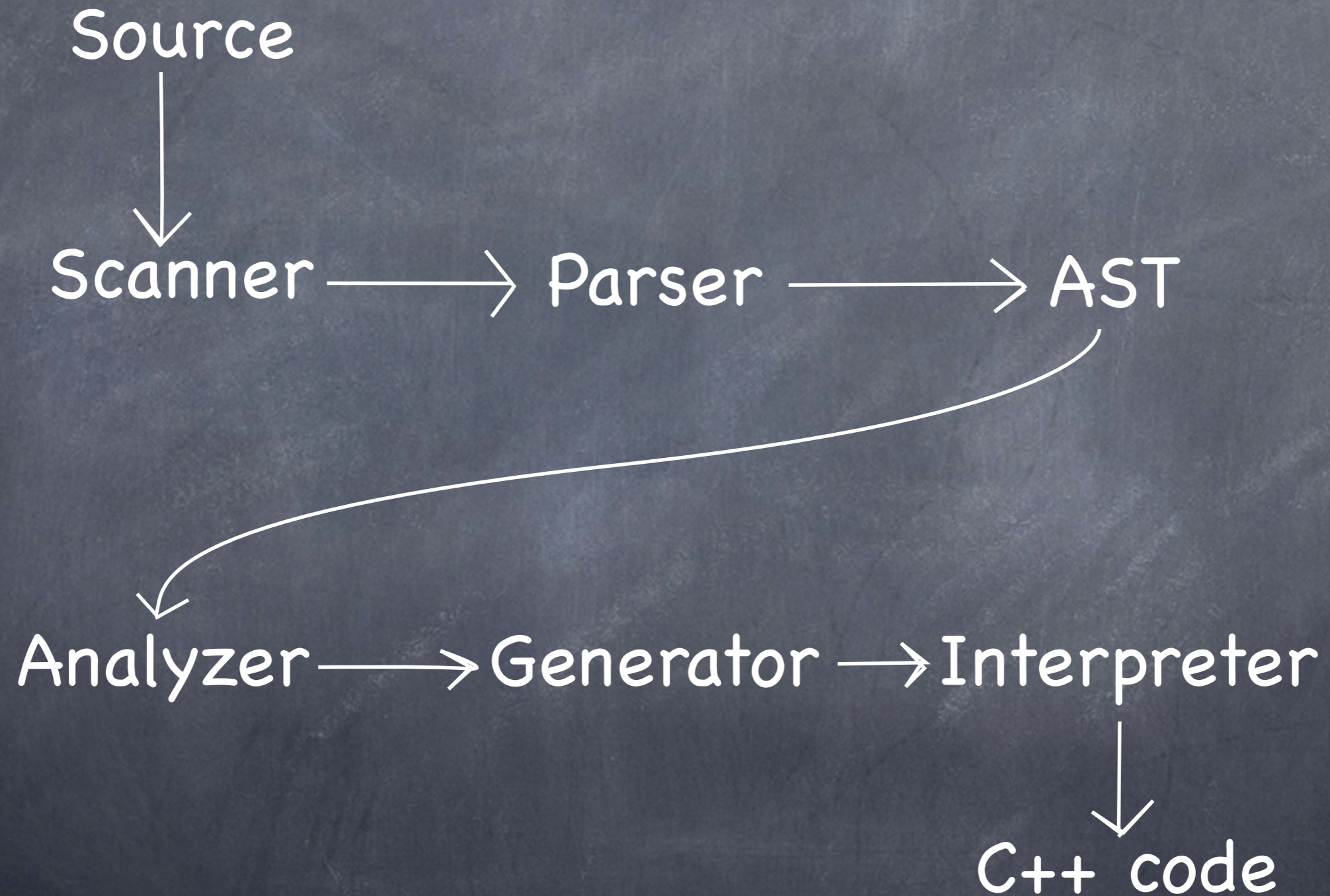
        if(c3!=c)
            __movePoint(p3);
            __leftClickPoint(p3);
            c3=c;
        end if
        if(c4!=c)
            __movePoint(p4);
            __leftClickPoint(p4);
            c4=c;
        end if
        if(c5!=c)
            __movePoint(p5);
            __leftClickPoint(p5);
            c5=c;
        end if
        if(c6!=c)
            __movePoint(p6);
            __leftClickPoint(p6);
            c6=c;
        end if
```

Tutorial -- A program for playing whack a mole

```
    if(c7!=c)
        __movePoint(p7);
        __leftClickPoint(p7);
        c7=c;
    end if
    if(c8!=c)
        __movePoint(p8);
        __leftClickPoint(p8);
        c8=c;
    end if
    if(c9!=c)
        __movePoint(p9);
        __leftClickPoint(p9);
        c9=c;
    end if
    __delay(300);
end while

return 0;
end function
```

Architecture



Lessons

- Team work
- Compromising, especially different opinion
- Design is important
- Ask immediately
- Test case should be recorded
- Start early

Conclusion

• Compiler

analyzer.ml	428
ast.mli	45
bytecode.mli	81
parser.mly	157
sast.mli	39
scanner.mli	132
scanner._test.ml	74
type.mli	26

• Interpreter

point.cpp	18
instructions.cpp	23
rgb.cpp	20
source.cpp	227
tyname.cpp	13
winapi.cpp	55
Program.cpp	935

• Total: 2273

Thank you