

# *EasKey Language*

*---A New Keyboard/Mouse Entry Era*

Keqiu Hu (kh2567)  
Jinqi Huang (jh3297)  
Xiaoyu Huang (xh2185)  
Zongheng Wang (zw2223)  
Lizhong Zhang(lz2324)

## **I. Background and Motivation**

Technology makes people's life easier. Computer is supposed to help people complete iterative and boring calculations. However, with the high-speed development of software industry, software installed in a computer becomes more and more complicated and often requires thousands of mouse clicks and keyboard inputs, most of which are repetitive and boring.

Take a few examples, in the course registration period, you may wish to register a class that is already full. So the most common way may be that you constantly press the “Class Search” button to refresh the website in order to see that if someone has just dropped the course so you are lucky enough to fill in the position. Or perhaps you need to update a database system with thousands of operations that are similar to each other, so you have to input all the instructions one by one. Or you may want to click your mouse as fast as possible to win a game which needs immediate reaction.

In a circumstance like any of these, it is definitely a waste of time to do finish the task in normal ways presented above. So, is there something we can do to make it easier and faster and more efficient? If what we do is simply a combination of keyboard inputs and mouse clicks, why don't we let the computer itself to finish the task? These results in our idea to design a language to facilitate people doing iterative keyboard and mouse input. And the solution is our EasKey Language!

## **II Introduction**

EasKey aims to facilitate programming of keyboard and mouse actions. It consists of most fundamental operations and functions of regular programming languages, such as branches and loops. Besides, all the potential keyboard and mouse actions are realized by EasKey internal functions and integrated tightly with the compiler, which makes the process of keyboard and mouse programming super easy for users.

EasKey could solve three kinds of problems. First, people sometimes encounter

repetitive operations of keyboard and mouse when using computers, such as document processing or monitoring data online, e.g. course registration status in Columbia University. Then they can write corresponding programs based on their operations and the feedbacks from the screen to ease the work. Second, human beings sometimes cannot operate keyboard and mouse fast enough as needed, especially for some complex and quick operations in certain games. For this reason, people could set desired sequences and running time in the program ahead of time, and when the operations are needed, just run the program. Third, not all software supports customizable shortcut keys. Therefore, flexibility is restricted tremendously in the user experience. At this time, EasKey could easily make a new set of shortcut keys in user's favor.

In sum, the purpose of EasKey is to provide with great convenience and reliability both in the coding and application process.

### **III. Scope**

The EasKey Language is introduced to help us deal with our daily computer operations conveniently and make the interaction between humans and computers more transparent. We can use this language to record and describe the motion or the input of mouse or keyboard, respectively. So more specifically, what can this language do? Or what kind of people would use this language to do what kind of things? Let's see:

#### **Game player:**

Have you ever been tired of keeping clicking the Mouse when playing certain games (like World of Warcraft), just for some routine operation? With this language, you don't need to spend the whole day in front of the Computer just to do those boring operations. You can tell the computer how to do it automatically all by itself and spare some time to have a cup of coffee.

#### **Student:**

Students always complain about course registrations, because they cannot get what they want. With this language, you can tell the computer how to register a course, when and which button to click to register course. You just need to turn on a computer and tell it what should be done, and then go to school, leaving the computer keeping refreshing the system and clicking the "ADD" button until the course has vacancy.

#### **Office Worker**

With no doubt, office workers may have to deal with hundreds of boring routine operation every day, like receiving emails, dealing with Excel files or other documents. With this language, you can just tell the computer how to do these stuffs.

#### **Crazy Apple Fans**

Hah, using this language, fans don't need to wait till the late night just to buy an iPhone 5. They can just write a little script with this language, compile and run it an hour or so before the sell starts, then you can just leave the browser unattended and the compiled program can help you place a order instantly when they start selling it.

#### **IV. Language Specification**

##### *a. Primitive Data types*

*point*: is a data type which helps to define a point of screen or window.

*cursor*: is a data type which helps to easily find the location of cursor in the screen.

*color*: is a data type which defines a kind of color

*window*: is a data type which helps to define window ,helps to get attributes of window.

*location*: has two values, x coordinate and y coordinate

We also support the basic data types listed below :

*int* : this is an integer

*float* : this is a floating point number

*string* : this is a sequence of characters

*bool*: true or false

##### *b. Comment*

# this is a comment

Use sigle line comment with # prefix

##### *c. Declarations*

The way to declare a variable is similar with C or C++ language. Before a variable can be used, it must be declared.

variable type variable\_name;

variable type variable\_name = variable\_value;

for example:

int temp; #declare a variable named temp

int temp=0; #declare a variable named temp and its value is 0.

Point point; #declare a point

##### *d. Operators*

Our language will support the following operators.

a + b	Addition
a - b	Subtraction
a * b	Multiplication
a / b	Division
a > b	Greater than
a < b	Less than
a == b	Equals
a >= b	Greater than or equal to
a <= b	Less than or equal to

#### *e. Control-Flow*

The syntax for program control flow are given below:

```
If [ expr ]  
#do something  
elif [ expr ]  
#do something  
Else  
#do something  
endif
```

if the boolean expression is true, then code in the subsequent block is executed and if it is false, judge the value of next expression. If it is true, then code in the subsequent block is executed. If it is false, else block is executed.

```
while [boolean_expr] {  
    #do something  
}
```

For as long as the boolean expression is true, the code enclosed in the curly braces is executed.

```
do{  
    # do something  
}while [ boolean_expr ]
```

This is similar to the while loop, it executes code enclosed in the curly braces first, and then judges Boolean expression.

#### *f.Function*

A function may have some inputs and outputs. It may need several arguments and return value.

```
return_type func function_name(type arg1,type arg2){  
    #do something  
}
```

This function has two arguments and need to return output.

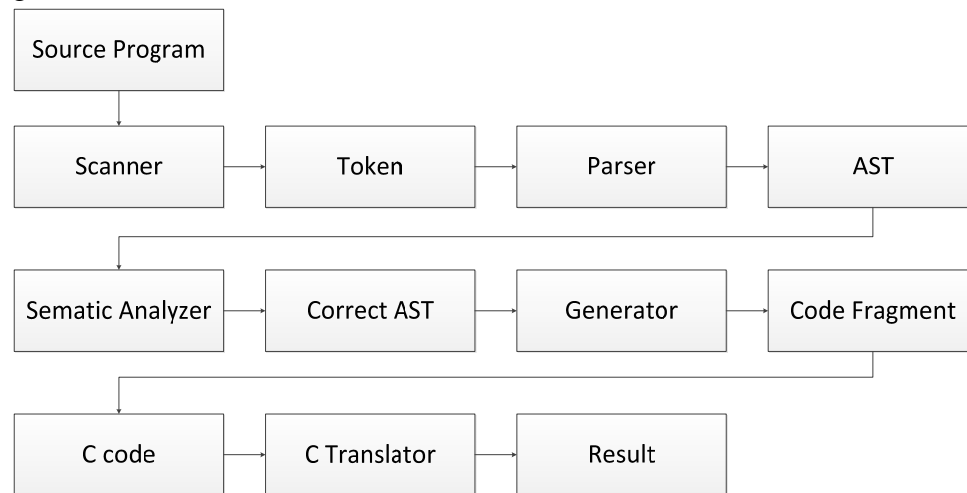
EasKey helps users to capture mouse and keyboard input easily. There are some specific functions.

```
Clic(int x,int y) #this is imitate the click of a button  
doubleClic(int x,int y) #this is imitate the double click of a button  
getCursor() #get the location of cursor  
getLocation() #get the location of point
```

#### **IV. Implementation**

After the design of our language, the source code would be compiled to C language. By using C libraries, we will get our final results.

The whole process of compiles is shown in Fig1. First, use our language to write program in order to do what you want to do. Source program becomes tokens after scanner. Then tokens are explained to be Abstract Syntax Tree. After semantic Analyzer and Generator, source code will be explained to C code. Use C library, we can finally get our result.



(Fig 1 Compiling Process)

Program would be compiled and executed on Windows. By using Window API, It is easy for us to capture mouse and keyboard input.

At the beginning of the program, we will use windows library.

```
#include <Windows.h>
```

It will be easier and more convenient to help you complete your program which needs to capture mouse and keyboard input by using our language.

#### **V. Example**

In the previous course registration example, a most common way to register the course is to enter the course name into the textbox, and then click the “class search” button thousands of times to see if you are lucky enough to coincide with other people's dropping that course. As seen from this example, the process likes:

- Step1: [click “class search” button]
- Step2: [determine whether the course if FULL]
- Step3: [if the ADD COURSE button appear, click it]
- Step4: [otherwise repeat Step1 through Step 3]

So what if we have the following EasKey language script:

```
color addcourse="#000000";  
cursor search="80,60";  
cursor add="100,120" //just an example position of the ADD COURSE button  
while(add.color!=addcourse)  
{  
    click(search);  
}  
click(add);
```

After compilation, the program will automatically run and make your course registration process a most painless way.