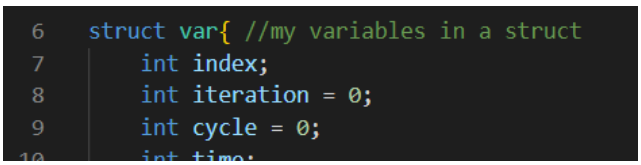
AutoClicker Technical Documentation

Structure of my code:

For my code, I chose in a first step to use a struct variable instead of using class. Personally, it was better to use a struct because the program was small enough to do without a class and for me there was a better understanding for this method.



And for this method, I only need 1 .cpp file for all my project.

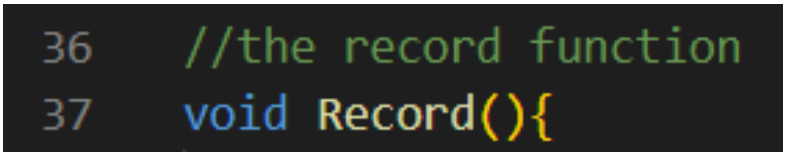
Choice of my functions:

I chose not to use too many functions, I kept to the basics for this project.

My main functions are:

-The “Record” Function:

This allows you to record your cycle of clicks



-The “Click” Function:

This allows you to execute your cycle of clicks and you can configure how many cycles you want, or you can configure it to run indefinitely.

```
71   //a preview of the record
72   void Preview(int r1[], int r2[], int l1[], int l2[], int sleep)
```

-The “Preview” Function:

This allows you to have a preview of your record, it’s only move the mouse where you click on your record. So don’t worry, this function do not click.

```
71   //a preview of the record
72   void Preview(int r1[], int r2[], int l1[], int l2[], int sleep)
```

-The “Main” Function:

It run the entire program and display the menu for beginning to record your clicks

For the “Main” Function, I also show you how I built my menu

```
119  //the menu
120  int main(){
121      std::cout<<"MENU AutoClicker"<<std::endl;
122      std::cout<<"\n"<<std::endl;
123      std::cout<<"Start Record with NUM1"<<std::endl;
124      std::cout<<"Stop Record with NUM2"<<std::endl;
125      std::cout<<"Start the autoclicker with NUM3"<<std::endl;
126      std::cout<<"Stop the autoclicker with NUM4"<<std::endl;
127      std::cout<<"Preview the record with NUM5"<<std::endl;
128      std::cout<<"Display your record with NUM6"<<std::endl;
129      std::cout<<"Delete your record with NUM7"<<std::endl;
130      std::cout<<"Display the menu with NUM8"<<std::endl;
131      std::cout<<"Export the record with NUM9"<<std::endl;
132      std::cout<<"\n"<<std::endl;
133      std::cout<<"Please use the pad only for the menu"<<std::endl;
```

But you can also see how the menu display in the console in the User Documentation.

A view of the most interesting parts of these functions:

-For the “Record” Function:

This is how I recover the user click (this part is only for the right mouse click and the wheel mouse click).

```
POINT mouse;
while(clicker.cond1){
    if(key(VK_RBUTTON)||key(VK_MBUTTON)){
        GetCursorPos(&mouse);
        x1[clicker.index] = mouse.x;
        y1[clicker.index] = mouse.y;
        x2[clicker.index] =  y2[clicker.index] = 0;
```

-For the “Click” Function:

This is how I move my cursor to the coordinates of the click in the record (this part is only for clicking with the right mouse click).

```
SetCursorPos(r1[clicker.x], r2[clicker.x]);
mouse_event(MOUSEEVENTF_RIGHTDOWN, 0, 0, 0, 0);
mouse_event(MOUSEEVENTF_RIGHTUP, 0, 0, 0, 0);
```

-For the “Preview” Function:

There is nothing much to say, it’s similar to the “Click” Function but without the mouse event.

-For the “Main” Function:

Display the record history:

```
    std::cout<<"Name of your task :"<<clicker.name<<std::endl;
    for(int j=0; j<clicker.index; ++j){
        std::cout<<"Click right:"<<x2[j]<<","<<y2[j]<<std::endl;
        std::cout<<"Click left:"<<x1[j]<<","<<y1[j]<<std::endl;
```

-The export feature:

It writes the clicks of the record in the .txt file in the folder of my project.

New entries overwrite old ones.

```
std::string filename("export.txt");
std::ofstream test(filename.c_str());
if(test){
    std::string txt;
    for(int i = 0 ; i < clicker.index ;i++)
        txt += convertInt(x2[i])+" "+convertInt(y2[i])+" "+convertInt(x1[i])+" "+convertInt(y1[i])+"\n";
    test << txt << std::endl ;
```

You can rename the .txt file but you must modify the code after.
Otherwise, it won't work anymore.

-The execution of the record:

-For a define number of cycles:

```
for(int g=0; g<clicker.cycle; ++g){
    if (!clicker.cond2){
        break;
    }
    Click(x1,y1,x2,y2,clicker.time);
    if (key(VK_NUMPAD4)){
        clicker.cond2 = false;
        std::cout << "You stop the clicker" << std::endl;
    }
```

-For an infinite number of cycles:

```
while(TRUE){
    if (!clicker.cond2){
        break;
    }
    Click(x1,y1,x2,y2,clicker.time);
    if (key(VK_NUMPAD4)){
        clicker.cond2 = false;
        std::cout << "You stop the clicker" << std::endl;
    }
```