

Today in CS161

- **Week #5** *****weekly visit D2L*****
 - ***Practicing...***
 - *Review from Reading*
 - *If statements*
 - *Loops*
 - **Continue writing Programs**
 - Using if statements and loops
 - **Graphics**
 - Begin creating the tic tac toe program

Shackelford Reading

- **Explain the difference between data types and variables**
- **What kind of data type should we use for**
 - **Keeping track of how many students are in your class**
 - **Storing if you are a freshman, sophomore, junior, or senior**
 - **Storing your GPA**

Shackelford Reading

- **Now, pick variable names to represent these...**
 - **Keeping track of how many students are in your class**
 - **Storing if you are a freshman, sophomore, junior, or senior**
 - **Storing your GPA**
- **Give an example of a poor variable name**

Shackelford Reading

- **Write an algorithm that will...**
 - **Prompt the user to type in three numbers and then read in those three numbers**
 - **Calculate the largest of the three and display the result**
- **Give an example of an algorithm that would require multiple if, else statements**

Review of C++ from Lectures

- Show an example of how to do **output**
- Show an example of how to do **input**
- Write a C++ statement to display a message ***Please enter your age:*** to the console (screen).
- What does every ***complete statement*** end with?
- Why do we need to use ***#include <iostream>*** ?

Review of C++ from Lectures

- Which of the following are NOT valid assignment statements:
 - **Total = 9;**
 - **72 = amount;**
 - **Profit == 129;**
 - **Letter = 'w';**
 - **Testing = 100**

Review of C++ from Lectures

- Which of the following are NOT valid output statements:
 - `cout << "Hello CS161!";`
 - `cout << "I said: "Have Fun!" ok?";`
 - `cout < 10;`
 - `cout << Programming is great fun!;`

Review of C++ from Lectures

- **Write a program that displays**
 - Your name
 - City that you live in
 - College major
- **Write C++ code** that calculates the number of acres in a tract of land with 389,767 square ft (*one acre of land has 43,560 square ft*)

Answer

```
#include <iostream>
using namespace std;

int main()
{
    float acres; //number of acres in a tract of land

    acres = 389767 / 43560;
    cout <<" There are " <<acres <<" acres in 389,767 sq ft" <<endl;

    cin.get(); //pause
    return 0;
}
```

Answer...more general

```
#include <iostream>
using namespace std;

const int SQFT_ACRE = 43560;

int main()
{
    float acres; //number of acres in a tract of land
    float sqft; //how many square ft you have

    cout << "How large is your tract - in square ft? ";
    cin >> sqft;  cin.get();
    acres = sqft / SQFT_ACRE;

    cout << " There are " << acres << " acres in " << sqft << " sq ft" << endl;

    cin.get(); //pause
    return 0;
}
```

Practicing Loops

- **Write C++ code** that calculates the average rainfall for three months. Ask the user to enter the amount of rainfall for June, July, and August in inches. The program should display the average rainfall from that data received
 - *Try this without a loop*
 - *Now, let's see what this is like with a loop!*

Answer...program fragment

```
float average; //this will hold the answer
float june_rain, july_rain, aug_rain; //rainfalls in inches

cout << "Please enter the rainfall in inches for June: ";
cin >> june_rain;  cin.get();

cout << "Please enter the rainfall in inches for July: ";
cin >> july_rain;  cin.get();

cout << "Please enter the rainfall in inches for August: ";
cin >> aug_rain;  cin.get();

//calculate the average
average = (june_rain + july_rain + aug_rain)/3.0;

cout << "The average rainfall was: " << average << "in" << endl;

cin.get(); //pause
```

Answer...with loops!!

```
float average; //this will hold the answer
float rain;    //rainfalls in inches
int counter = 1;
float total = 0; //a running total

do
{
    cout << "Please enter the rainfall in inches for month " << counter << " : ";
    cin >> rain;  cin.get();

    ++ counter;
    total += rain; //we need a running total
} while (counter <= 3);

//calculate the average
average = total / 3.0;

cout << "The average rainfall was: " << average << "in" << endl;
cin.get(); //pause
```

Answer...more general

```
float average; //this will hold the answer
float rain;    //rainfalls in inches
int counter = 1;
float total = 0; //a running total
int num_months = 0; //number of months to average

cout << "How many months of rainfall are we averaging? ";
cin >> num_months; cin.get();
do
{
    cout << "Please enter the rainfall in inches for month " << counter << " : ";
    cin >> rain;  cin.get();

    ++ counter;
    total += rain; //we need a running total
} while (counter <= num_months);
//calculate the average
average = total / num_months;

cout << "The average rainfall was: " << average << "in" << endl;
```

Answer...(I'd use a for loop)

```
float average; //this will hold the answer
float rain;    //rainfalls in inches
float total = 0; //a running total
int num_months = 0; //number of months to average

cout << "How many months of rainfall are we averaging? ";
cin >> num_months; cin.get();

for (int counter = 1; counter <= num_months; ++ counter)
{
    cout << "Please enter the rainfall in inches for month " <<counter <<" : ";
    cin >> rain;  cin.get();

    total += rain; //we need a running total
}

//calculate the average
average = total / num_months;

cout << "The average rainfall was: " <<average <<"in" <<endl;
```

Adding if statements

- **What if in the previous program,**
 - ***The user entered a negative number?***

```
cout << "How many months of rainfall are we averaging? ";  
cin >> num_months; cin.get();
```

```
if (num_months <= 0)  
    cout << "A negative (or zero) was received...no good!";  
else  
{
```


Giving the user another chance

- **Now change that to give the user another chance!,**
 - ***The user entered a negative number?***

```
cout << "How many months of rainfall are we averaging? ";  
cin >> num_months; cin.get();
```

```
while (num_months <= 0)  
{  
    cout << "A negative was received...no good!";  
    cout << endl << "Try again!" << endl;  
    cout << "How many months of rainfall are we averaging? ";  
    cin >> num_months; cin.get();  
}
```

Giving the user another chance

- **What about with a do-while?**
 - ***The user entered a negative number?***

```
do
{
    cout << "How many months of rainfall are we averaging? ";
    cin >> num_months; cin.get();
    if (num_months <= 0)
        cout << "A negative was received...no good" <<endl <<"Try Again!"
            << endl;
}
while (num_months <= 0);
```

Beginning the Tic Tac Toe Program

1. Algorithm: Display the board

- 1. Find out what the window size is**
 - 1. Let's keep the width and height the same**
- 2. Set the color to white**
- 3. Set the line width to wider, so we can see the board**
- 4. Draw 2 vertical lines $1/3^{\text{rd}}$ and $2/3^{\text{rd}}$ across the window**
- 5. Draw 2 horizontal lines $1/3^{\text{rd}}$ and $2/3^{\text{rd}}$ down the window**

2. Now think about what variables you will need

- 1. Whole number to keep track of the window size**

Beginning the Tic Tac Toe Program

```
int window_size;

cout << "Please select the size of your window: ";
cin >> window_size; cin.get();

initwindow(window_size,window_size);

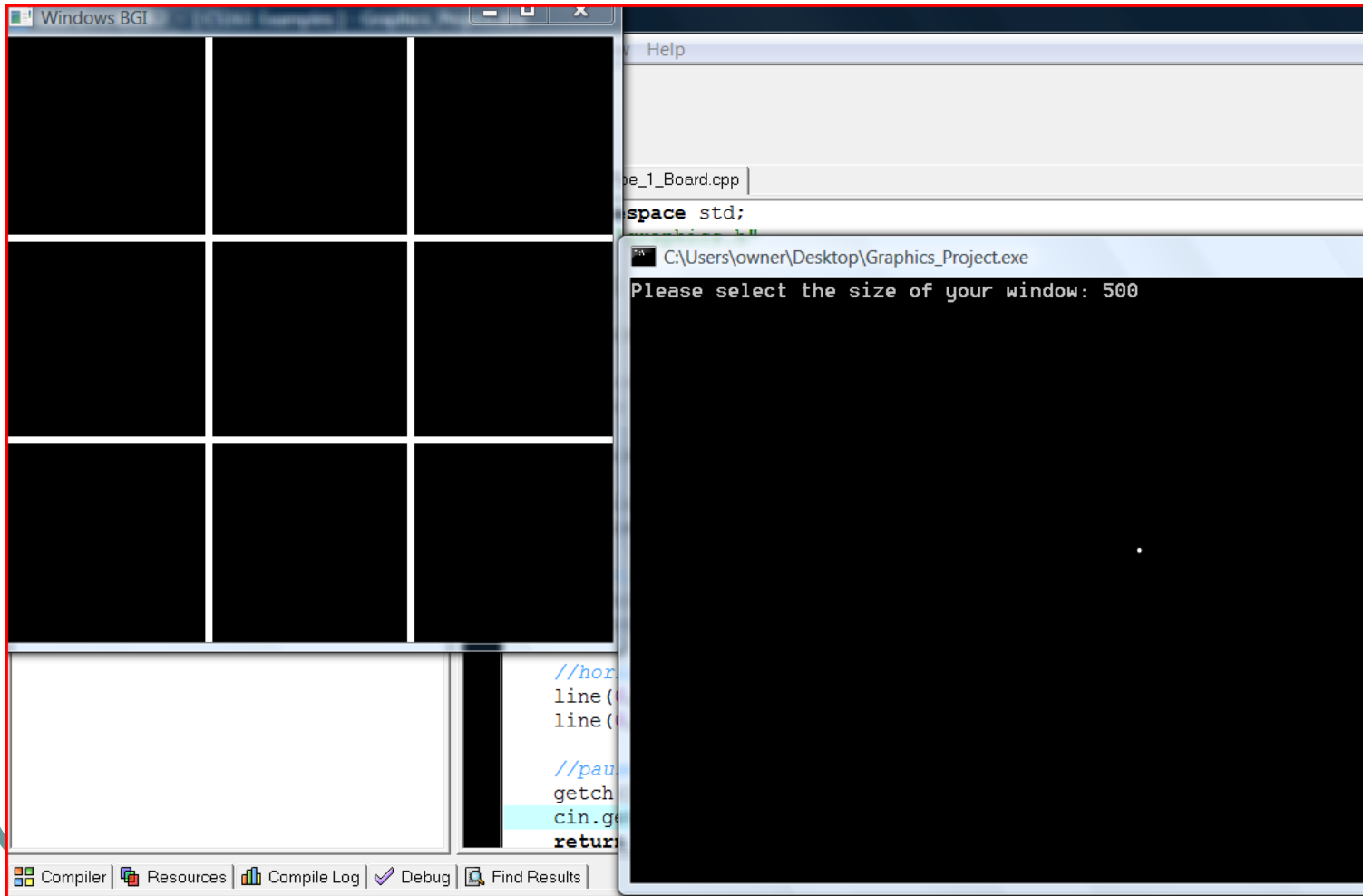
setcolor(15); //15 is WHITE
setlinestyle(0,0,6); //Solid, No patter, 6 is VERY wide

//vertical lines
line(window_size/3, 0, window_size/3, window_size);
line(window_size*2/3, 0, window_size*2/3, window_size);

//horizontal lines
line(0,window_size/3, window_size, window_size/3);
line(0,window_size*2/3, window_size, window_size*2/3);

//pause
getch(); //for graphics window
cin.get(); //for console window
return 0;
}
```

Tic Tac Toe Board...



Next Step...

1. Algorithm: Select Player

- 1. Ask the user who will be the first to play**
 - 1. X or O**
- 2. If the user enters a lower case or some other character**
 - 1. Display an error message**
 - 2. Give them another chance**
- 3. Echo the choice to the user, once it is correct**

2. Now think about what variables you will need

- 1. User Input**
 - 1. Character data**
 - 2. Must be a capital X or capital O**

Selecting the Player

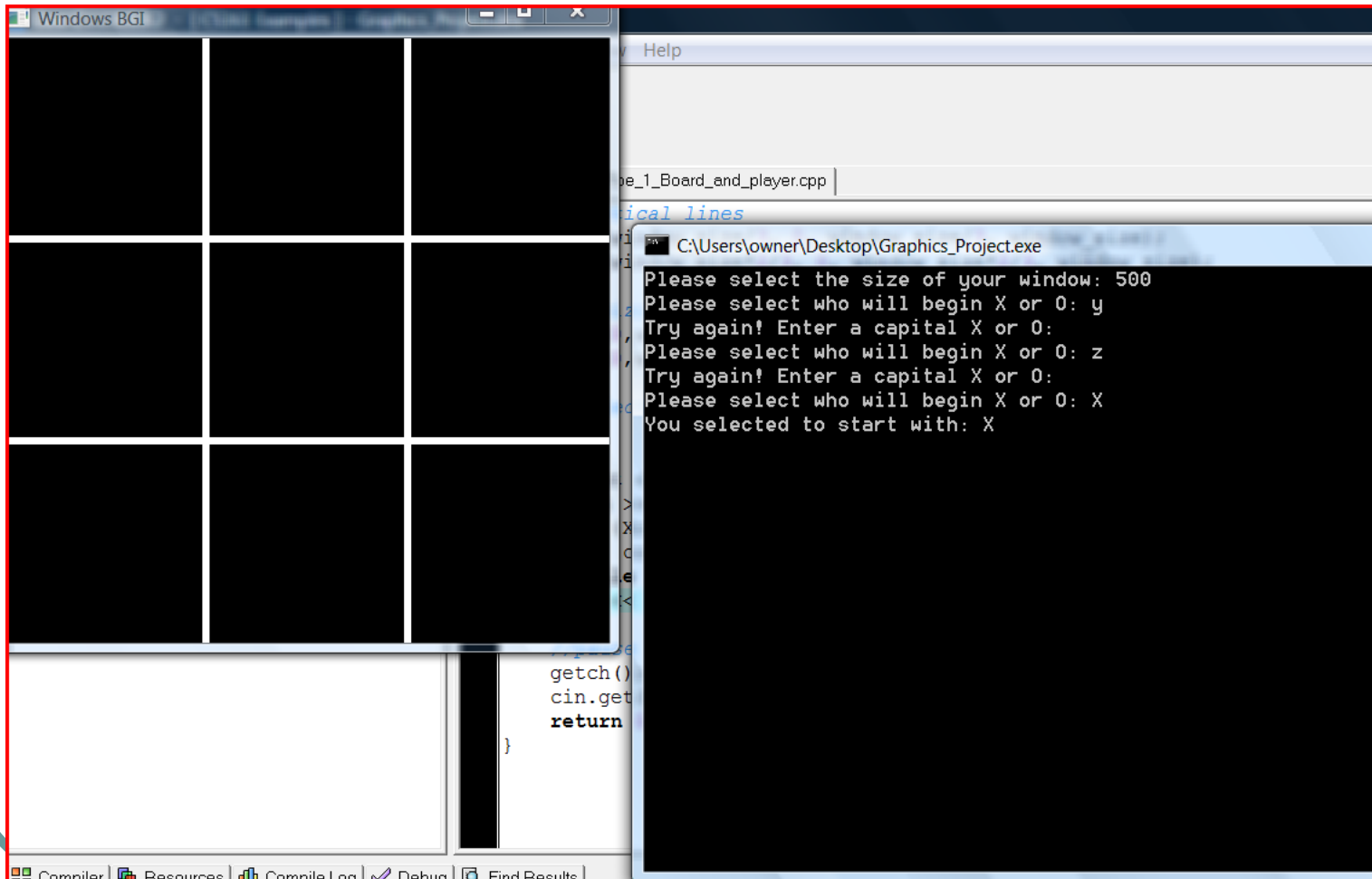
```
//Select the starting player
char XorO;
do
{
    cout << "Please select who will begin X or O: ";
    cin >> XorO;      cin.get();

    //check for error conditions
    if (XorO != 'X' && XorO != 'O')
        cout << "Try again! Enter a capital X or O: " << endl;

} while (XorO != 'X' && XorO != 'O');

//echo the selection
cout << "You selected to start with: " << XorO << endl;
```

Selecting the Player...



Selecting a location on the board

1. Algorithm: Select Location on the board

- 1. Prompt the user to select a location on the board**
- 2. Receive input from the mouse in the graphics window**
 - 1. Get the x,y location where the mouse hit occurred**
 - 2. Find out which square the x,y hit happened in**
 - 1. If x is in the first 1/3rd of window**
 - 1. And y is in the first 1/3rd of window**
 - 1. UPPER LEFT corner**
 - 2. Else if y is in first 2/3rd of window**
 - 1. MIDDLE LEFT**
 - 3. Else**
 - 1. LOWER LEFT corner**

Selecting a location on the board

2. Else if x is in first $2/3^{\text{rd}}$ of window
 1. And y is in the first $1/3^{\text{rd}}$ of window
 1. UPPER MIDDLE
 2. Else if y is in first $2/3^{\text{rd}}$ of window
 1. MIDDLE MIDDLE
 3. Else
 1. LOWER MIDDLE
3. Else
 1. And y is in the first $1/3^{\text{rd}}$ of window
 1. UPPER RIGHT
 2. Else if y is in first $2/3^{\text{rd}}$ of window
 1. MIDDLE RIGHT
 3. Else
 1. LOWER RIGHT

Mouse Hit!

```
cout << "RIGHT BUTTON: Please select the location on the board with the mouse: " << endl;
while (!ismouseclick(516));           //wait for a mouse click...RIGHT button
int x = mousex();                     //this is where the mouse click happened (x,y)
int y = mousey();
setcolor(YELLOW);                     //color of the text
settextstyle(0,0,5);                  //create a really large character (5)
settextjustify(1,1);                  //center both horizontally and vertically

int square_size = window_size/3;     //I created two new variables to help with the calculations
int half_size = square_size/2;

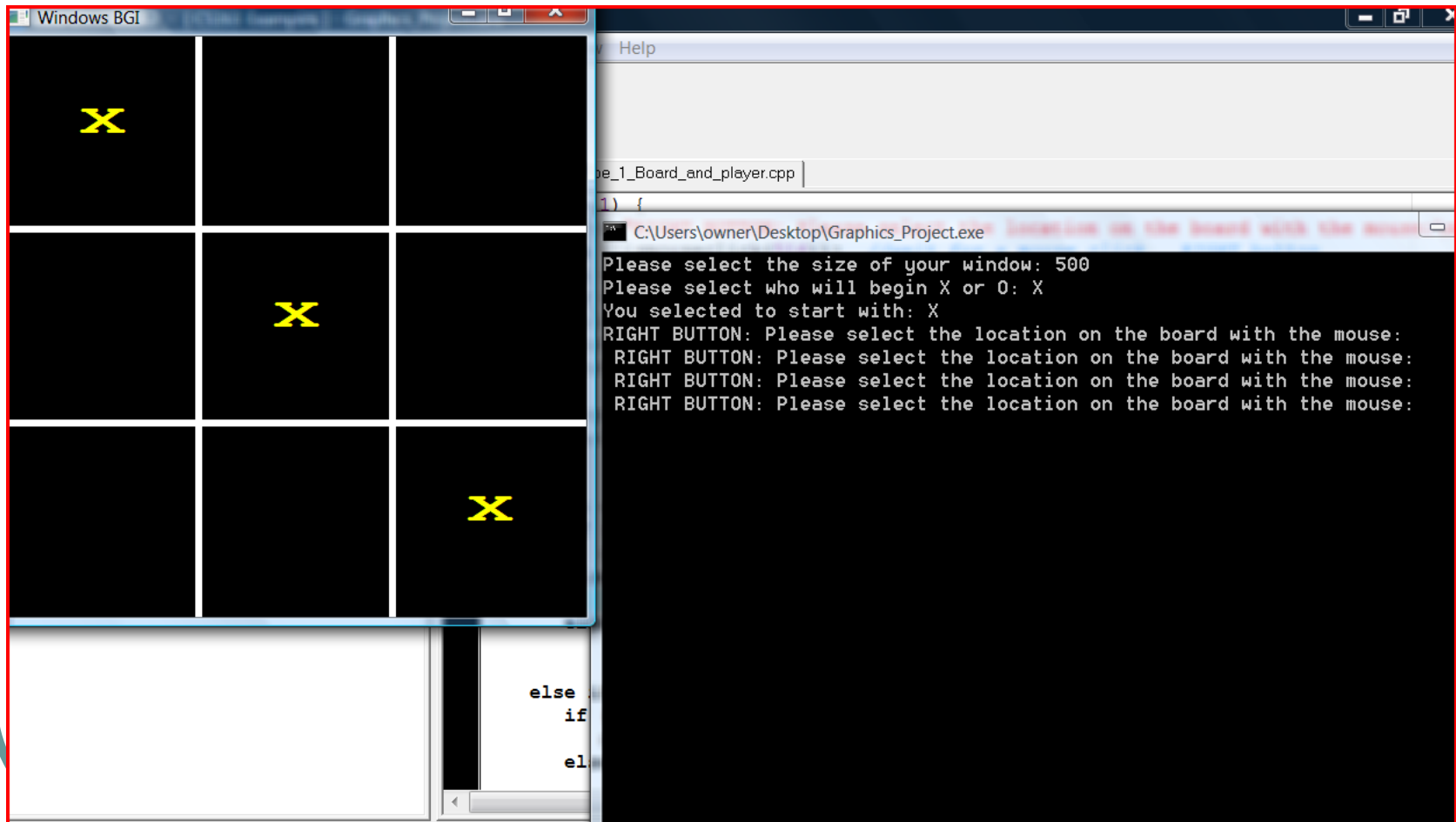
if (x < window_size/3)                //is it in the left side of the board?
    if (y < window_size/3)
        outtextxy(half_size,half_size,"X");
    else if (y < window_size*2/3)
        outtextxy(half_size>window_size*2/3-half_size,"X");
    else
        outtextxy(half_size>window_size-half_size,"X");
...//plus more
```

Mouse Hit!

```
else if (x < window_size * 2/3) //middle column
    if (y < window_size/3)
        outtextxy(window_size * 2/3-half_size,half_size,"X");
    else if (y < window_size*2/3)
        outtextxy(window_size * 2/3-half_size,window_size*2/3-half_size,"X");
    else
        outtextxy(window_size * 2/3-half_size,window_size-half_size,"X");

else
{
    //right hand column
    if (y < window_size/3)
        outtextxy(window_size-half_size,half_size,"X");
    else if (y < window_size*2/3)
        outtextxy(window_size-half_size,window_size*2/3-half_size,"X");
    else
        outtextxy(window_size-half_size,window_size-half_size,"X");
}
clearmouseclick(516);    //important!
```

Testing it out so far...in a loop



We have just begun...

1. We still need to...

- 1. Display the appropriate X versus O, depending on who the player is**
- 2. Add a loop to continue until there is a ..**
 - 1. Winner or**
 - 2. Cat scratch**
- 3. All to be done....next lecture!**