IIT Bombay

Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Elementary Graphics Events
Guest Lecturer: Dr. Abhiram Ranade

Quick recap

IIT Bombay

Quick recap

IIT Bombay

Last session: Coordinate based graphics facility of Simplecpp.

Quick recap

IIT Bombay

Last session: Coordinate based graphics facility of Simplecpp.

This session: Handling graphical input

Quick recap

IIT Bombay

Last session: Coordinate based graphics facility of Simplecpp.

This session: Handling graphical input

Section 5.5, 5.7, 6.4 of "An introduction to programming through C++", McGraw Hill Education, 2014.

Graphical input

IIT Bombay

Graphical input

IIT Bombay

Input given through the mouse, touch screens.

Graphical input

IIT Bombay

Input given through the mouse, touch screens.

By graphical input we mean information provided to a program by clicking on a screen using a mouse or using a touch screen.

Graphical input

IIT Bombay

Input given through the mouse, touch screens.

By graphical input we mean information provided to a program by clicking on a screen using a mouse or using a touch screen.
The information usually includes the coordinates of the point at which a click was made, in addition to what was clicked/how.

Graphical input

IIT Bombay

Input given through the mouse, touch screens.

By graphical input we mean information provided to a program by clicking on a screen using a mouse or using a touch screen.
The information usually includes the coordinates of the point at which a click was made, in addition to what was clicked/how.

Advantages:

Graphical input

IIT Bombay

Input given through the mouse, touch screens.

By graphical input we mean information provided to a program by clicking on a screen using a mouse or using a touch screen.
The information usually includes the coordinates of the point at which a click was made, in addition to what was clicked/how.

Advantages:

- Some data is inherently geometric, e.g. coordinates of the vertices of a polygon. Better given by clicking.

Graphical input

IIT Bombay

Input given through the mouse, touch screens.

By graphical input we mean information provided to a program by clicking on a screen using a mouse or using a touch screen.
The information usually includes the coordinates of the point at which a click was made, in addition to what was clicked/how.

Advantages:

- Some data is inherently geometric, e.g. coordinates of the vertices of a polygon. Better given by clicking.
- User interfaces, in which buttons can be clicked are very convenient.

Graphical input

IIT Bombay

Input given through the mouse, touch screens.

By graphical input we mean information provided to a program by clicking on a screen using a mouse or using a touch screen.
The information usually includes the coordinates of the point at which a click was made, in addition to what was clicked/how.

Advantages:

- Some data is inherently geometric, e.g. coordinates of the vertices of a polygon. Better given by clicking.
- User interfaces, in which buttons can be clicked are very convenient.

This lecture: Graphical input involving mouse clicks.

Graphical input

IIT Bombay

Input given through the mouse, touch screens.

By graphical input we mean information provided to a program by clicking on a screen using a mouse or using a touch screen.
The information usually includes the coordinates of the point at which a click was made, in addition to what was clicked/how.

Advantages:

- Some data is inherently geometric, e.g. coordinates of the vertices of a polygon. Better given by clicking.
- User interfaces, in which buttons can be clicked are very convenient.

This lecture: Graphical input involving mouse clicks.

Later: other details such as dragging, mouse buttons..

Handling mouse clicks: Function `getClick()`

IIT Bombay

Handling mouse clicks: Function `getClick()`

IIT Bombay

Signature:

```
int getClick();
```

Handling mouse clicks: Function `getClick()`

IIT Bombay

Signature:

```
int getClick();
```

Causes the program to wait for the user to click the mouse.

Handling mouse clicks: Function `getClick()`

IIT Bombay

Signature:
`int getClick();`

Causes the program to wait for the user to click the mouse. When the user clicks, at some position `(x,y)`, the function returns. The value returned is `65536*x+y`.

Handling mouse clicks: Function `getClick()`

IIT Bombay

Signature:
`int getClick();`

Causes the program to wait for the user to click the mouse. When the user clicks, at some position `(x,y)`, the function returns. The value returned is `65536*x+y`.
Here is how you will wait for a click and print the click coordinates.

Handling mouse clicks: Function `getClick()`

IIT Bombay

Signature:
`int getClick();`

Causes the program to wait for the user to click the mouse.
When the user clicks, at some position `(x,y)`, the function returns. The value returned is `65536*x+y`.
Here is how you will wait for a click and print the click coordinates.

```
int v = getClick();
cout <<"x:  " << v/65536 <<", y:  "<< v % 65536 <<
endl;
```

Handling mouse clicks: Function `getClick()`

IIT Bombay

Signature:
`int getClick();`

Causes the program to wait for the user to click the mouse. When the user clicks, at some position `(x,y)`, the function returns. The value returned is `65536*x+y`.
Here is how you will wait for a click and print the click coordinates.

```
int v = getClick();
cout <<"x:  " << v/65536 <<", y:  "<< v % 65536 <<
endl;
```

This works because the `x,y` coordinates are much smaller than 65536, and because `v` is an `int`. Thus integer division gets us back the integer quotient and the remainder on dividing by 65536.

Demo 1: Best fit line

IIT Bombay

Demo 1: Best fit line

IIT Bombay

Input: Points $(x_1, y_1), \ldots, (x_n, y_n)$ in the plane.

Demo 1: Best fit line

IIT Bombay

Input: Points $(x_1, y_1), \ldots, (x_n, y_n)$ in the plane.
Output: Line which fits them best.

Demo 1: Best fit line

IIT Bombay

Input: Points $(x_1, y_1), \ldots, (x_n, y_n)$ in the plane.
Output: Line which fits them best.

Standard algorithm: Find line $y = mx + c$ such that the vertical distance of the points to the line is minimized (in a least square manner).

Demo 1: Best fit line

IIT Bombay

Input: Points $(x_1, y_1), \ldots, (x_n, y_n)$ in the plane.
Output: Line which fits them best.

Standard algorithm: Find line $y = mx + c$ such that the vertical distance of the points to the line is minimized (in a least square manner).

Section 5.7 discusses how to calculate m, c.

Demo 1: Best fit line

IIT Bombay

Input: Points $(x_1, y_1), \ldots, (x_n, y_n)$ in the plane.
Output: Line which fits them best.

Standard algorithm: Find line $y = mx + c$ such that the vertical distance of the points to the line is minimized (in a least square manner).

Section 5.7 discusses how to calculate m, c.

Observation: It is more natural to give the points by clicking on the screen, and show the line by plotting it on the screen.

Demo 1: Best fit line

IIT Bombay

Input: Points $(x_1, y_1), \ldots, (x_n, y_n)$ in the plane.
Output: Line which fits them best.

Standard algorithm: Find line $y = mx + c$ such that the vertical distance of the points to the line is minimized (in a least square manner).

Section 5.7 discusses how to calculate m, c.

Observation: It is more natural to give the points by clicking on the screen, and show the line by plotting it on the screen.

Program gets points by clicking. Places a circle at the clicked position to mark it. Then plots the best-fit line.

Demo 2: Button based turtle controller

IIT Bombay

Demo 2: Button based turtle controller

IIT Bombay

In the first session on graphics, we studied Turtle graphics.

Demo 2: Button based turtle controller

IIT Bombay

In the first session on graphics, we studied Turtle graphics.

The turtle was controlled through the program, using commands forward, left, right.

Demo 2: Button based turtle controller

IIT Bombay

In the first session on graphics, we studied Turtle graphics.

The turtle was controlled through the program, using commands forward, left, right.

We show a new way of controlling the turtle: the user can click a "button" on the canvas to move the turtle forward, another to turn the turtle.

Demo 2: Button based turtle controller

IIT Bombay

In the first session on graphics, we studied Turtle graphics.

The turtle was controlled through the program, using commands forward, left, right.

We show a new way of controlling the turtle: the user can click a "button" on the canvas to move the turtle forward, another to turn the turtle.

A button on the canvas is merely a rectangle with text inside it.

Demo 2: Button based turtle controller

IIT Bombay

In the first session on graphics, we studied Turtle graphics.

The turtle was controlled through the program, using commands forward, left, right.

We show a new way of controlling the turtle: the user can click a "button" on the canvas to move the turtle forward, another to turn the turtle.

A button on the canvas is merely a rectangle with text inside it.

To check if a button has been clicked, we merely check if the click coordinates lie inside the rectangle.

Summary

IIT Bombay

Summary

IIT Bombay

We discussed elementary graphical input as supported in Simplecpp.

Summary

IIT Bombay

We discussed elementary graphical input as supported in Simplecpp.

- In many applications, data is graphical. So graphical input is useful, e.g. fitting line to points.

Summary

IIT Bombay

We discussed elementary graphical input as supported in Simplecpp.

- In many applications, data is graphical. So graphical input is useful, e.g. fitting line to points.
- User interfaces involving buttons/icons are very common. They can be easily implemented using `getClick`.

Summary

IIT Bombay

We discussed elementary graphical input as supported in Simplecpp.

- In many applications, data is graphical. So graphical input is useful, e.g. fitting line to points.
- User interfaces involving buttons/icons are very common. They can be easily implemented using `getClick`.
- Above situations may arise in many programming projects; graphical input will thus be very useful.