COMP1021
Introduction to Computer Science

Turtle Window Events

Gibson Lam, David Rossiter and Leo Tsui

Outcomes

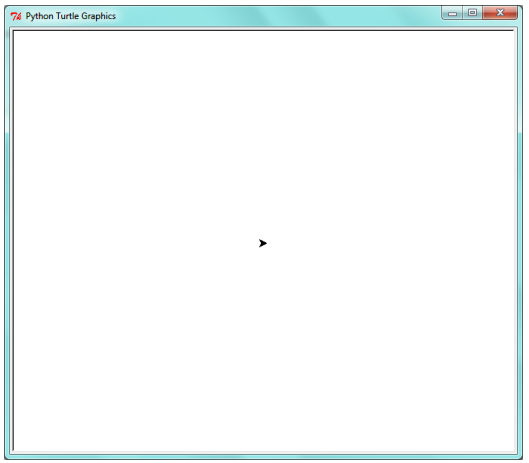
- After completing this presentation, you are expected to be able to:
  1. Explain what turtle window events are and how to handle them
  2. Write code to handle mouse click events
  3. Write code to handle mouse drag events

Using Text Input

- In a text-based program the user interacts with the program by entering text only
- You have already learned how to get text input using the `input()` function in Python

```
>>> age = input("How old are you? ")
How old are you? 7
>>> print(age)
7
```

Graphical User Interface



- When you use turtle graphics programming, apart from the usual text input/output, you also have a visual component, the turtle window
- The turtle window is the *graphical user interface* (GUI) of a turtle graphics program
- With the GUI, you can have many different kinds of interactions with the user, instead of just entering text

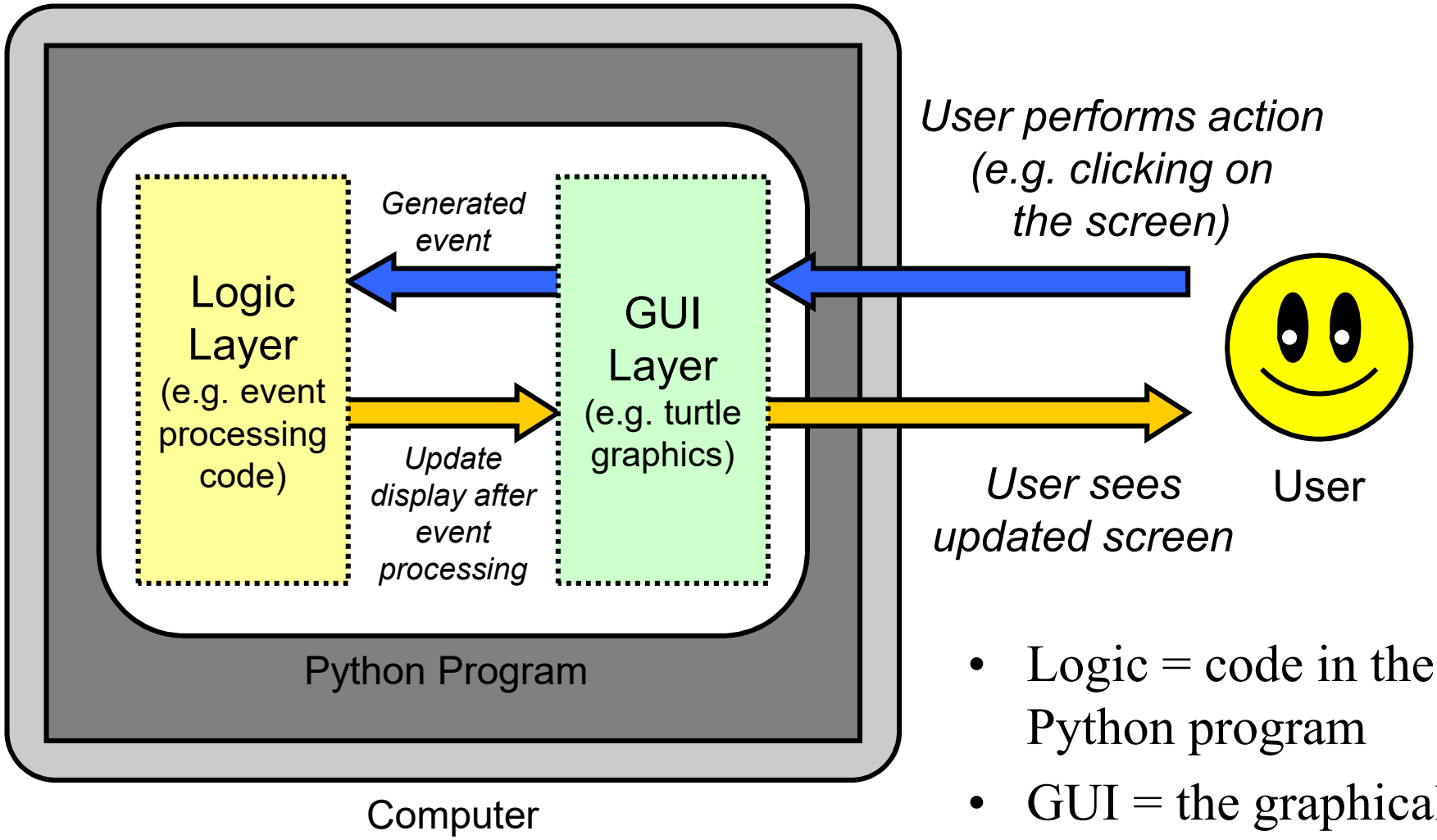
Events

- The interactions with a GUI are typically some actions performed by the user
- Here are a couple of examples:
  - When the user clicks inside the window
  - When the user presses a key
- These actions generate *events* in GUI systems
- An event can sometimes be generated by the system itself, such as a turtle timer event

Event Handling

- When a user performs a certain action it ‘triggers’ the corresponding event in the system
  - For example, if a user clicks on the turtle screen it will trigger a ‘mouse click’ event
- You can write code to handle a specific event so that when that event is triggered the corresponding code will be executed
  - For example, you can write code to show the current mouse position on the screen when there is a ‘mouse click’ event

Flow of Event Processing



- Logic = code in the Python program
- GUI = the graphical user interface

Turtle Graphics and Event Handling

- In turtle graphics programming you write *event handling functions* to handle events
- An event handling function is a Python function containing the code you want to run when a certain event is generated
- Depending on the events that you want to handle the event handling function uses a certain set of input parameters
- Sometimes we call an event handling function an *event handler*

Using an Event Handling Function

- An event handling function is just a normal Python function with some parameters
- When you want to use your function to handle an event you need to explicitly tell Python to do that
- We call this ‘assigning an event handling function to an event’
- You need to finish assigning event handling functions before running `turtle.done()`

```
...
Assigning event
handling functions
...

turtle.done()
```

Assigning a Function to an Event

- This is how you assign a Python function to an event:

```
turtle.on(event_name, event_handling_function)
```

The name of the event the Python function is being assigned to (e.g. click, timer, etc)

The Python function which will handle the event

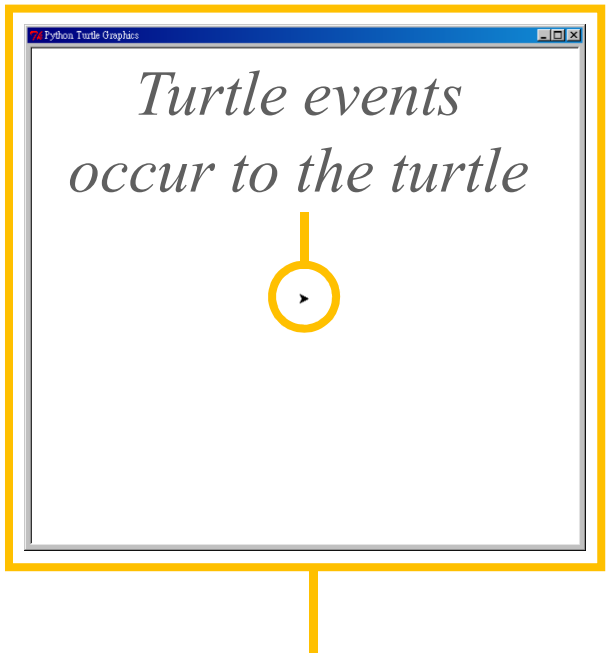
- In some situations, such as for keyboard events, one or more additional parameters are required:

```
turtle.on event_name (event_handling_function, ...)
```

Additional parameters go here

Event Types in Turtle Graphics

- For turtle graphics programming there are two types of event:
  - Turtle events
    - Events occur to the turtle ➤
  - Screen events
    - Events occur to the turtle window



Turtle Events

- Let's look at the following turtle events:
  1. The Mouse Click Event
     - This event is generated when a user clicks on a turtle
  2. The Mouse Drag Event
     - This event is generated when a user clicks and drags a turtle

1. The Mouse Click Event

- The `onclick()` function assigns a function to handle the mouse click event of a turtle
- For example:

```
def myclickfunc(x, y):
    . . .
```

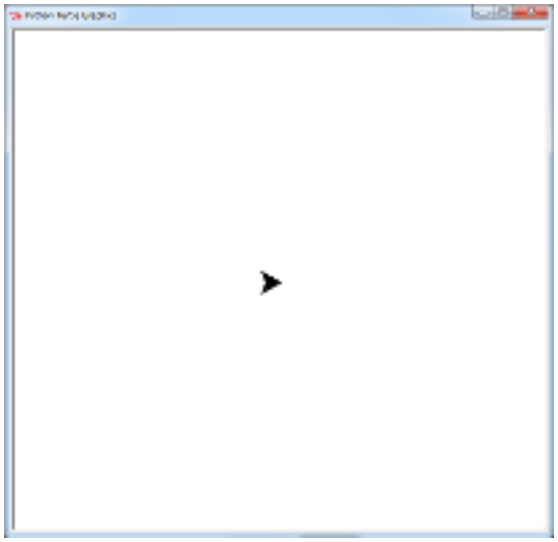
The x and y position of the mouse pointer are automatically given to the function

```
turtle_instance.onclick( myclickfunc )
```

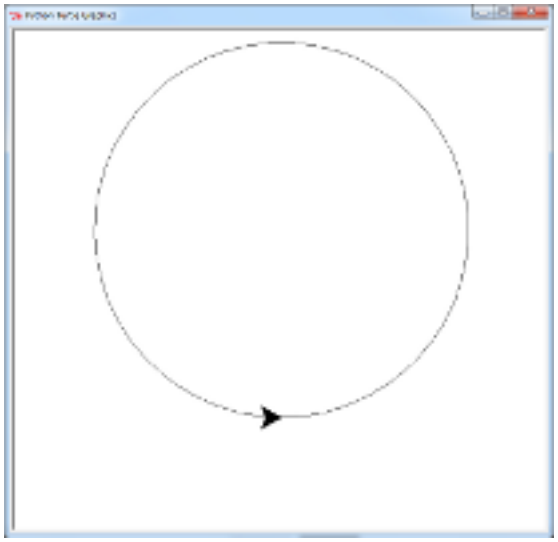
We are setting up a click event handling function for this particular turtle

The `myclickfunc` *function is assigned to handle the click event of the turtle*

Mouse Click Event Example 1/2



The user clicks on the turtle



The turtle doesn't do anything when the program begins

A circle is drawn after the user clicks on the turtle

Mouse Click Event Example 2/2

```
def drawcircle(x, y):
  turtle.up()
  turtle.goto(0, -180)
  turtle.down()
  turtle.circle(250)
```

This is the event handling function; it simply draws a circle when it is called

```
# Assign the function drawcircle to
# the turtle mouse click event
turtle.onclick(drawcircle)
```

The event handling function is assigned to this turtle

The `drawcircle` *function is assigned to handle the click event of the turtle*

2. The Mouse Drag Event

- The `ondrag()` function assigns a function to handle the mouse drag event of a turtle
- For example:

```
def mydragfunc(x, y):
    . . .
```

The x and y position of the mouse pointer are automatically given to the function

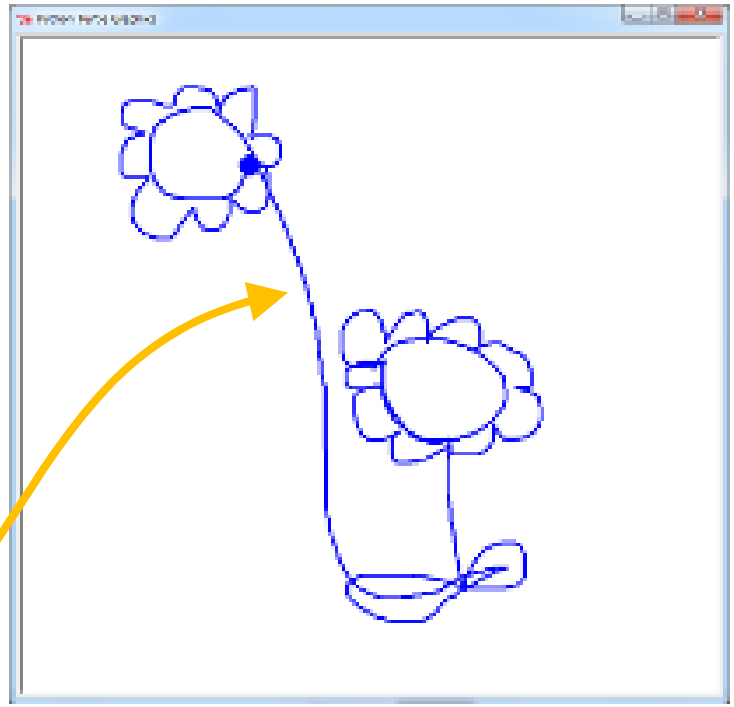
```
turtle_instance.ondrag( mydragfunc )
```

We are setting up a drag event handling function for this particular turtle

The `mydragfunc` *function is assigned to handle the drag event*

Mouse Drag Event Example 1/2

- In this example you can draw things by dragging the turtle
- For this program you cannot tell the turtle to stop drawing
- That means the drawings are connected in a big long line
  - For example, the sun and the flower on the right are connected by a line



Mouse Drag Event Example 2/2

- In this example, no user-defined function is used as the event handling function
- Instead, `turtle.goto(x, y)`, which is an existing turtle function, is used as the event handling function
- The event handling function is setup like this:

```
turtle.ondrag(turtle.goto)
```

Mouse Drag Event Example – Complete Code

```
import turtle

turtle.shape("circle")
turtle.color("blue")
turtle.pensize(3)

turtle.ondrag(turtle.goto)

turtle.speed(0)

turtle.done()
```

- The x and y position of the mouse pointer is automatically given to the `goto()` function
- So the turtle follows the movement of the mouse drag

Use the fastest turtle speed so that the drawings are shown almost instantly when the turtle is dragged