COMP1021
Introduction to Computer Science

# Clicking and Dragging Turtles

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



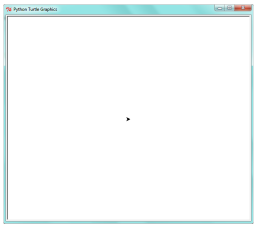
# Using Text Input

- In a text-based program the user interacts with the program through text only
- You have already learned text input and output using `input()` and `print()`

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



# Graphical User Interface



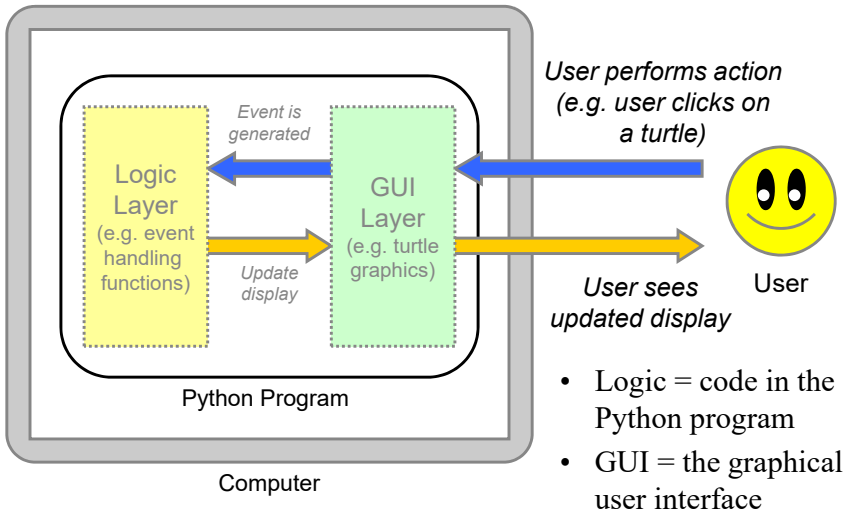
- When you use turtle graphics programming you have a visual component, the turtle window
- The turtle window is the *graphical user interface* (GUI) of a turtle graphics program
- With a GUI, you can have many different kinds of interactions with the program

# Event Handling

- When a user performs a certain action the corresponding event is generated
  - For example, if a user clicks on a turtle it will generate a 'click' event
- You can write code to handle the event
- For example, you can write code so that when there is a 'click' event the position of the turtle is shown on the screen



# Flow of Event Processing




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

# Turtle Graphics and Event Handling

- 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 particular event occurs
- Sometimes we simply call an event handling function an *event handler*



# Event Handling Functions

- You need to tell Python what function it should use to handle an event
- We say this is 'assigning an event handling function to an event'
- At the end of your program you need `turtle.done()`

```
import turtle
...
Assign event handling
functions to events
...
turtle.done()
```

- Basically, this means 'the turtle system has finished doing things'
- You must have this line of code when you do event handling

# Assigning a Function

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

*turtle_name*`.on(` *event_name* `,` *event_handling_function* `)`

*The name of the event the Python function is being assigned to (e.g. click, drag, also others)*

*This is the Python function which will be used to handle the event*



# Types of Event in Turtle Graphics

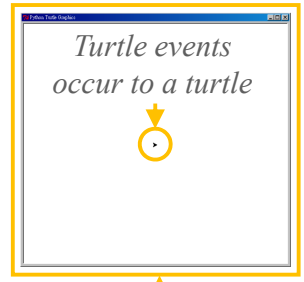
- For turtle graphics programming there are two types of event:

*We will look at these in this presentation*

- Turtle events
  - These events occur to a turtle ➤

*We may look at these in another presentation*

- Screen events
  - These events occur to the turtle window




*Screen 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 the user clicks on a turtle
  2. The Mouse Drag Event
     - This event is generated when the user clicks and drags a turtle



# 1. The Mouse Click Event

- The `onclick()` function assigns a function which does things when a turtle is clicked
- For example:

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

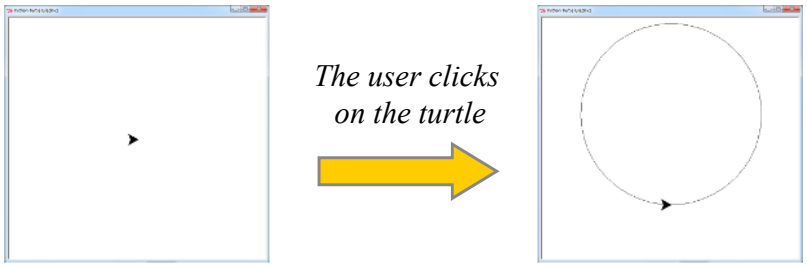
*The turtle system automatically gives the x and y values where the turtle was clicked to the function*

*turtle_name*`.onclick(` `myclickfunc` `)`

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

*The* `myclickfunc` *function will be executed when this turtle is clicked on*

# Click Event Example



*The user clicks on the turtle*

The turtle doesn't do anything when the program begins

After the user clicks on the turtle a circle is drawn



# Click Event Example

```
import turtle

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

turtle.onclick( drawcircle )

turtle.done() # The turtle system has finished
```

*The turtle system automatically gives the x and y values where the turtle was clicked to the function*

*This is the event handling function. It can do whatever you like. Here it simply draws a circle.*

*In this example we use the default turtle*

*The* `drawcircle` *function will be executed when the turtle is clicked on*

# 2. The Mouse Drag Event

- The `ondrag()` function assigns a function which does things when a turtle is dragged
- For example:

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

*The turtle system automatically gives the x and y values of the drag position to the function*

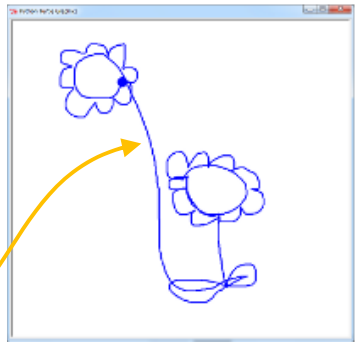
*turtle_name* `.ondrag(` `dragturtle` `)`

*The* `dragturtle` *function will be executed when this turtle is dragged*



# Mouse Drag Event Example

- 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 are connected by a line



# Drag Event Example 1

- This slide shows all the code

*In this example we use the default turtle*

*The turtle system automatically gives the x and y values of the position where the turtle was dragged to the function*

*This is the event handling function. It simply moves the turtle to the position where it is dragged.*

*Use the fastest turtle speed so the drawing is shown instantly*

*The* `dragturtle` *function will be executed when the turtle is dragged*

```
import turtle

def dragturtle(x, y):
    turtle.goto(x, y)

turtle.speed(0)
turtle.shape("circle")
turtle.color("blue")
turtle.pensize(3)
turtle.ondrag( dragturtle )
turtle.done() # The turtle system has finished
```

# Drag Event Example 2

- In the next example *turtle_name*`.goto(x, y)` (which you have used many times before) is used for the event handler function

*The x and y values of the turtle drag position are automatically given to turtle.goto()*

*the default turtle*

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



# Drag Event Example 2

- In this example we use the default turtle

- This slide shows all the code
- The x and y values of the turtle drag position are automatically given to `turtle.goto()`
- So the turtle follows the movement of the drag

*the default turtle*

*Use the fastest turtle speed so that the drawing is shown instantly when the turtle is dragged*

```
import turtle

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

turtle.speed(0)

turtle.done() # The turtle system has finished
```