Introduction to Computer Programming
Section 8

Lecture 16

Building an App
Interactive App Design

• Visual Components (View)
  – The code that draws the visual elements

• Knowledge Components (Model)
  – The code that implements what the visual elements represent
  – Global state, rules, meaning

• Interaction Components (Controller)
  – Users interacting with an app cause changes to “view” and the “model”
  – Interaction is thru keyboard, mouse, touch screen
Turtle Input/Output

• Print a string at the current turtle position

write("Hello", font=("Arial", 14, "normal"))

• Accept input from a dialog box

textinput("dialog box", "Enter name")
Take an action on Mouse Click

- Mouse interaction – your program can detect mouse movement and mouse clicks
- Example: close the turtle window if the user clicks anywhere on the canvas

```python
turtle.exitonclick()
```
Keyboard Interactions

- Bind program functions with keyboard keys

  ```python
  turtle.onkey("move_up", "Up")
  turtle.onkey("throw_dart", "t")
  ```

- “Listen” for keyboard events

  ```python
  turtle.listen()
  ```
Simulating A Game of Darts

• Problem can be decomposed:
  – Visual
    • Drawing a dartboard and darts
  – Interaction
    • “Throwing” a dart
  – Knowledge
    • How do you know whether a dart lands inside the bull’s eye, or what ring it’s in?
Drawing a dartboard

• Concentric circles, with center at (0,0)
  – Radius 50 pixels  (Bull’s eye)
  – Radius 100 pixels  (Inner ring)
  – Radius 150 pixels  (Outer ring)

• 400x400 outer square
Drawing a circle with radius R

Approximate a circle as a polygon with 360 sides. We can approximate the side length:

\[
\text{side\_length} = \frac{\text{Circumference}}{360}
\]

Since we know that Circumference = 2 \pi R

\[
\text{side\_length} = \frac{2\times\pi\times\text{radius}}{360}
\]

Example: If radius == 150, then

\[
\text{side\_length} = \frac{2\times\pi\times150}{360}
\]
def draw_circle(R) :
    circumference = 2 * math.pi * R
    side_length = circumference/360
    num_sides = 360
    polygon(num_sides, side_length)
Interaction: Throwing a Dart

• Binding a keyboard key to a function “throw_dart”

def throw_dart() :
    # Randomly pick an (x,y) coordinate
    x = random.randint(-200, 200)
    y = random.randint(-200, 200)
    ....
turtle.goto(x, y)
turtle.dot()   # Prints a dot on the canvas
Knowledge: Is \((X, Y)\) A Hit?

Need to Compute Distance to \((0, 0)\)
Code for Bull’s Eye

if  z < 50 :
    turtle.write(“Bulls Eye”)
elif z < 100 :
    turtle.write(“Inner Ring”)
elif z < 150 :
    turtle.write(“Outer Ring”)
else :
    turtle.write(“Miss!”)
Use The Pythagorean Theorem

\[ x^2 + y^2 = z^2 \]

\[ Z = \sqrt{x^2 + y^2} \]
What is $\pi$?

Area $= \pi r^2$

If the radius is 1, then

Area $= \pi$
The area of the circle is a percent of the area of the square.

Area of square == 4
Monte Carlo Simulation

Area = 4 (BlueDarts/TotalDarts)
Just Look at Upper Quadrant
The ratio is the same!

Area = 4 \text{ (BlueDarts/TotalDarts)}
Design

• Visual
  – Don’t know $\pi$, so can’t draw the concentric circles
  – The frame is 2 x 2

• Interaction
  – Throw darts: $x, y$ random numbers (0.0, 1.0)
  – Since need to throw lots of them, no interaction

• Knowledge
  – Know the radius is 1, so we can determine hits and misses (blue dots and red dots)
Adventure Game 2
Find the pot of gold in the vault

Cellar
Foyer
Parlor
Kitchen
Bedroom
Vault
Visual Components

- Draw a room
- Show bumping into a wall
- Show turtle path as the turtle changes rooms
Knowledge

The map of the rooms.

The turtle is always in the middle of a room.

Global variables:
- Room name, where the turtle is
- Flags indicating which rooms were visited
Interaction: Player selects a direction (N, S, E, W)

- Player selects a direction (N, S, E, W)
  - Is there a room?
    - NO: BUMP!!!!!!
    - YES: First visit?
      - YES: Draw room
        - Mark room visited
      - NO: The turtle walks to the room
Functions

Is there a room?

- BUMP!!!!!!

Enter the room

First visit?

- NO
  - Draw room
    - NO
      - Mark room visited
  - The turtle walks to the room

Mark room visited
Combine Entering Room Logic in A Function

- **BUMP!!!!!!**
- **Is there a room?**
  - **Enter the room**
  - **First visit?**
    - **Draw room**
    - **Mark room visited**
    - **The turtle walks to the room**
Implement The Room Map with Four Functions

- BUMP!!!!!!
- Move North
- Enter the room
- First visit?
- Draw room
- Mark room visited
- The turtle walks to the room
Function Hierarchy

Main program

- move_north()
- move_south()
- move_east()
- move_west()

- enter_room(direction, room)
- bump (direction)

- draw_room()
- first_visit()
- mark_visited()