OOP Design

Instructor: Paruj Ratanaworabhan

Bank Account

Consideration

- How can we represent a bank account? (state of an object)
- What are some operations associated with a bank account? (operations associated with an object)
- Where is a bank account stored?

Bank Account Representation

- The state of a bank account is determined by:
  - Account number as string
  - Account name as string
  - Account balance as float
  - Other attributes as QR code, phone number, etc.

```
class BankAccount:

    def __init__(self, number, name, balance):
        self.number = number
        self.name = name
        self.balance = balance

    # methods definitions to follow
```

Bank Account Operations

- Deposit
- Withdraw
- Transfer
- __str__ (to show the representation in string)

```
class BankAccount:

    def __init__(self, number, name, balance):
        self.number = number
        self.name = name
        self.balance = balance

    def deposit(self, amount):
        pass

    def withdraw(self, amount):
        pass

    def transfer(self, amount, to_account_num):
        pass

    def __str__(self):
        pass
```

Bank Account Database

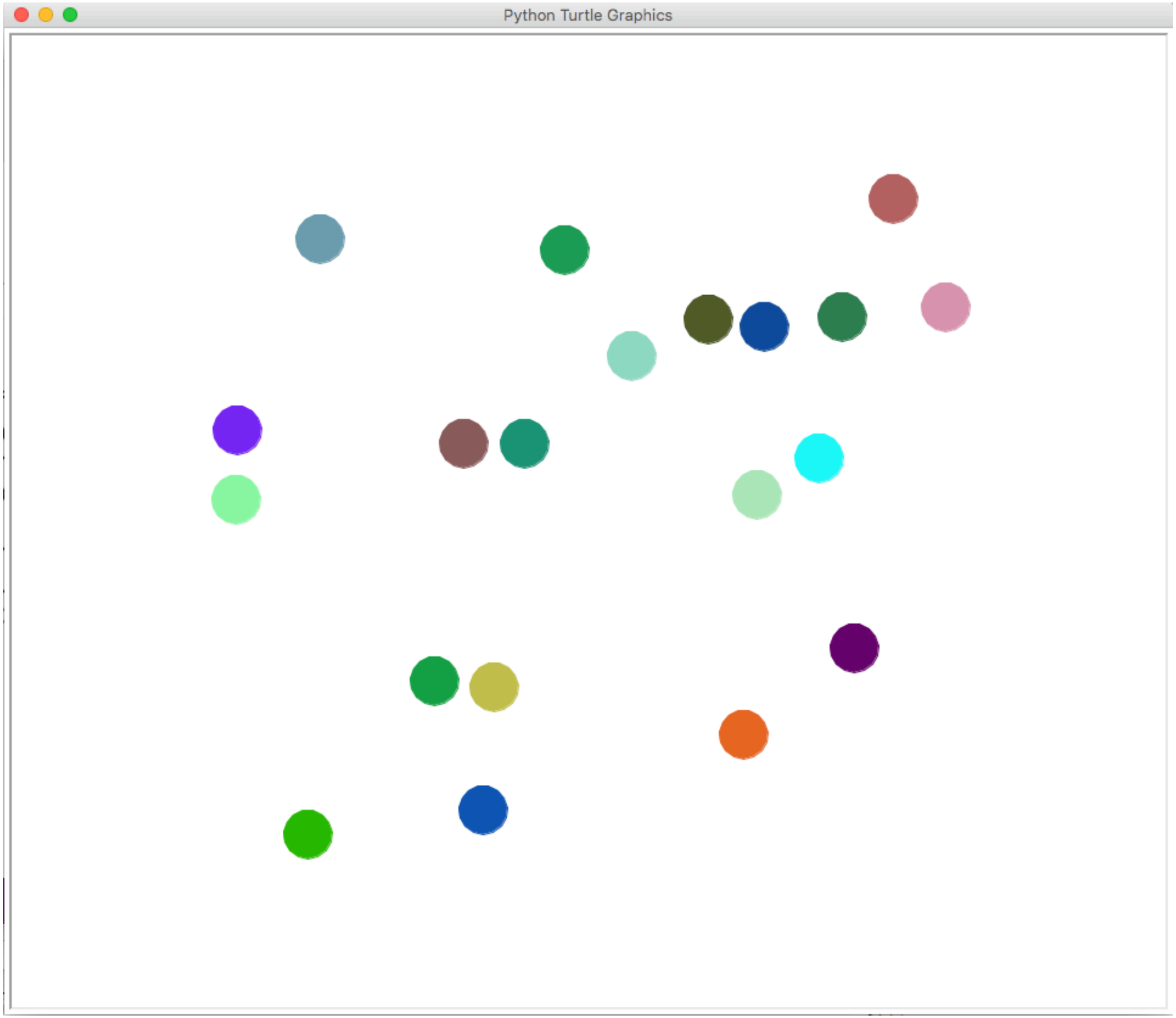
- When a bank account is created, it needs to be stored in a database
- Must have another class called BankAccountDB
- There will only be one database for all bank accounts
- The database is represented by a list of all bank accounts
- You can do the following operations on the database:
  - Search for a bank account
  - Insert a bank account to it
  - Delete a bank account from it

```
class BankAccountDB:

    def __init__(self):
        self.account_list = []

    def insert(self, bank_account):
        pass

    def search(self, account_number):
        pass

    def delete(self, account_number):
        pass

    def __str__(self):
        pass
```

Bank Account Running Code Example

```
account_database = BankAccountDB()
account1 = BankAccount(“123-45-001”, “Account1”, 500.00)
account_database.insert(account1)
account1 = BankAccount(“123-45-002”, “Account2”, 1000.00)
account_database.insert(account2)
search_result = account_database.search(“123-67-003”)
account1.deposit(200)
```

Ball Bouncing Simulation


Python Turtle Graphics

Consideration

- How can we represent a ball? (state of an object)
- What are some operations associated with a ball in this simulation context? (operations associated with an object)

Ball Representation

- The state of a ball is determined by:
  - Radius as float
  - Position as list of two float numbers, x and y
  - Velocity as list of two float numbers, vx and vy
  - Color as tuple of three RGB numbers from 0 to 255

```
class Ball:

    def __init__(self, radius, position, velocity, color):
        self.radius = radius
        self.position = position
        self.velocity = velocity
        self.color = color

    # methods definitions to follow
```

Ball Operations

- Draw
- Move

```
class Ball:

    def __init__(self, radius, position, velocity, color):
        self.radius = radius
        self.position = position
        self.velocity = velocity
        self.color = color

    def draw(self):
        pass

    def move(self):
        pass
```

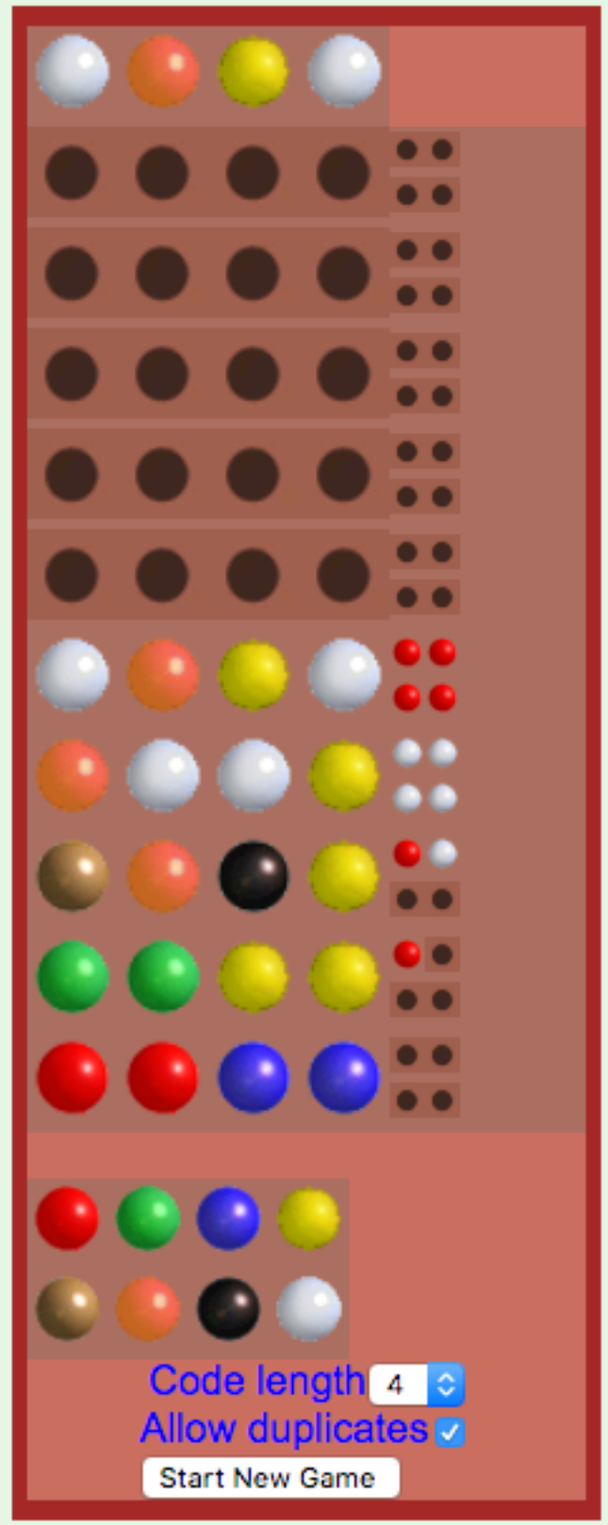
Ball Running Code Example

```
ball_A = Ball(5.0, [0, 0], [1, 2], (255, 255, 255))
ball_B = Ball(2.5, [20, -10], [-1, 3], (255, 0, 255))
while (True):
    turtle.clear()
    ball_A.draw()
    ball_B.draw()
    ball_A.move()
    ball_B.move()
    turtle.update()
```

Let’s Play Mastermind

https://www.webgamesonline.com/mastermind/

Play Mastermind Online



Our Setup

- We will only be using 6 colors represented by the number 1 to 6
- Duplicates are allowed; so there are 6 * 6 * 6 * 6 = 1296 possible puzzles
- The program accepts user inputs from a terminal, extracts the first four characters, and provides a clue whether an input is closer to solving the puzzle
- A star * indicates that there is a color that is positioned correctly
- A letter o indicates that there is a color that is positioned incorrectly
- The game is played on until a player successfully solves the puzzle

Sample Run 1

```
What is your guess?: 1122
Your guess is 1122
*oo

What is your guess?: 1213
Your guess is 1213
**oo

What is your guess?: 1231
Your guess is 1231
oooo

What is your guess?: 2113
Your guess is 2113
****

You solve it after 4 rounds
```

Sample Run 2

```
What is your guess?: 1122
Your guess is 1122


What is your guess?: 3344
Your guess is 3344
**

What is your guess?: 5566
Your guess is 5566
*

What is your guess?: 3363
Your guess is 3363
**oo

What is your guess?: 3336
Your guess is 3336
****

You solve it after 5 rounds
```

en.wikipedia.org/wiki/Mastermind_(board_game)

Consideration

- How can we represent a Mastermind board game? (state of an object)
- What are some operations associated with this game? (operations associated with an object)

In-class Exercise

- Design an OO program for this game

Solution Outline

```
class MasterMindBoard:

    def __int__(self):
        str = ''
        for i in range(4):
            import random
            str += random.randint(1, 6)
        self.solution = str
        self.num_guesses = 0
        self.guess = ''
        self.clue = ''

    def guess(self, input_guess):
        pass

    def display_clue(self):
        pass

    def done(self):
        pass

new_game = MasterMindBoard()
while (True):
    input_guess = input("What is your guess?: ")
    print('Your guess is', input_guess)
    MasterMindBoard.guess(input_guess)
    MasterMindBoard.display_clue()
    if MasterMindBoard.done():
        break
```