



จัดทำโดย

นางสาวณิชา ทองอยู่ สาขา IT รหัสนักศึกษา 6606021610087 ชั้นปีที่1

นางสาวกัญจนพร จุตรพรพรหม สาขา IT รหัสนักศึกษา 6606021610095 ชั้นปีที่1

นางสาววรินยุพา เจียมอยู่ สาขา IT รหัสนักศึกษา 6606021610176 ชั้นปีที่1

นางสาวเพ็ญภา คมะปะเต สาขา IT รหัสนักศึกษา 6606021610206 ชั้นปีที่1

ตอนเรียนที่ IT 1-RA

นางสาวพัชรภา คู่มเจริญ สาขา IT รหัสนักศึกษา 6606021611164 ชั้นปีที่1

ตอนเรียนที่ IT 1-RB

เสนอ

ผู้ช่วยศาสตราจารย์สมชัย เชียงพงศ์พันธ์

รายงานฉบับนี้เป็นส่วนหนึ่งของการเรียนวิชา 060243103 Problem Solving in IT

สาขาวิชา เทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ วิทยาเขต ปราชินบุรี

ภาคเรียนที่ 1 ปีการศึกษา 2566

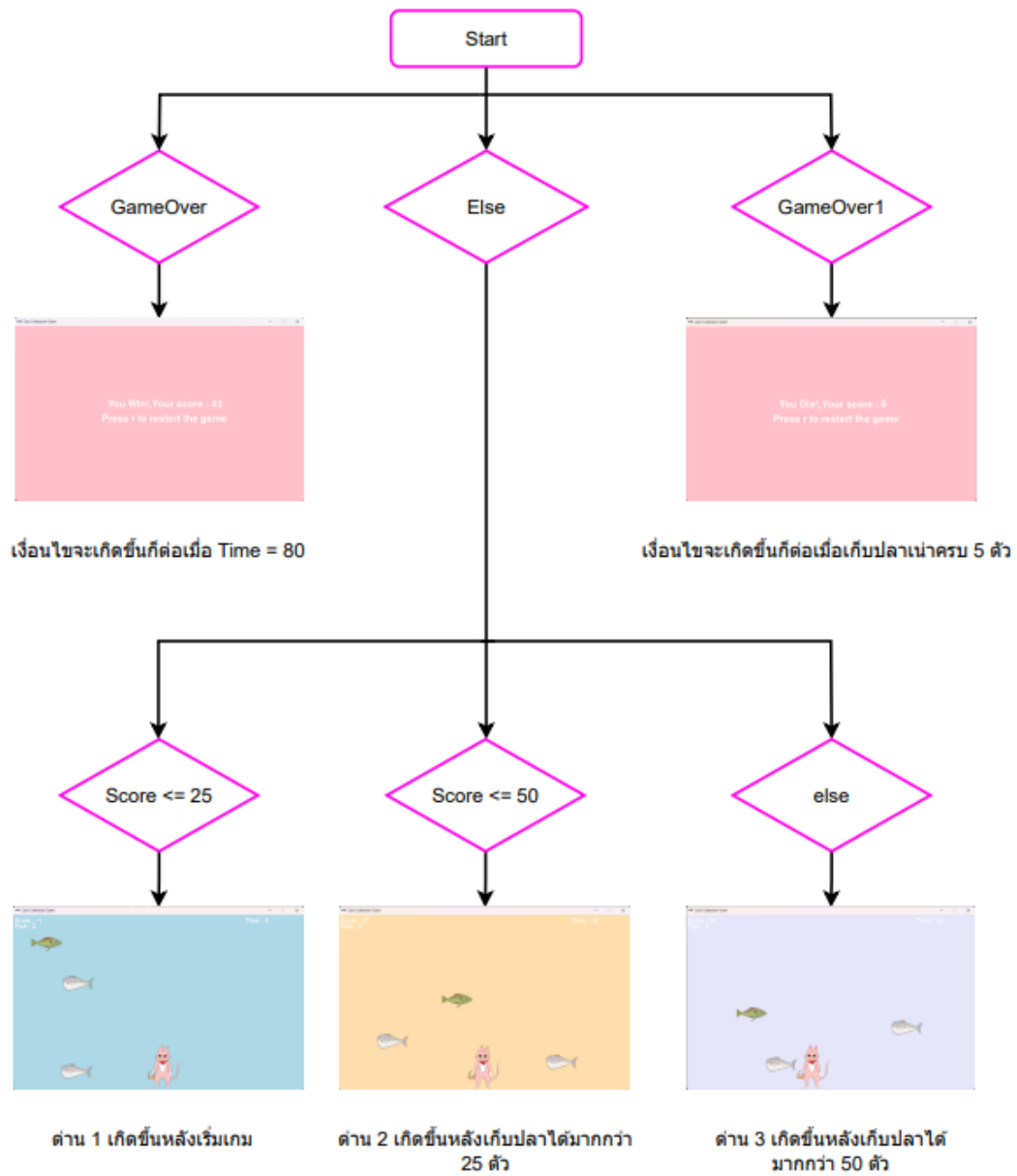
Python Zero

1.แนวคิดและหลักการเกี่ยวกับการพัฒนาโปรแกรม

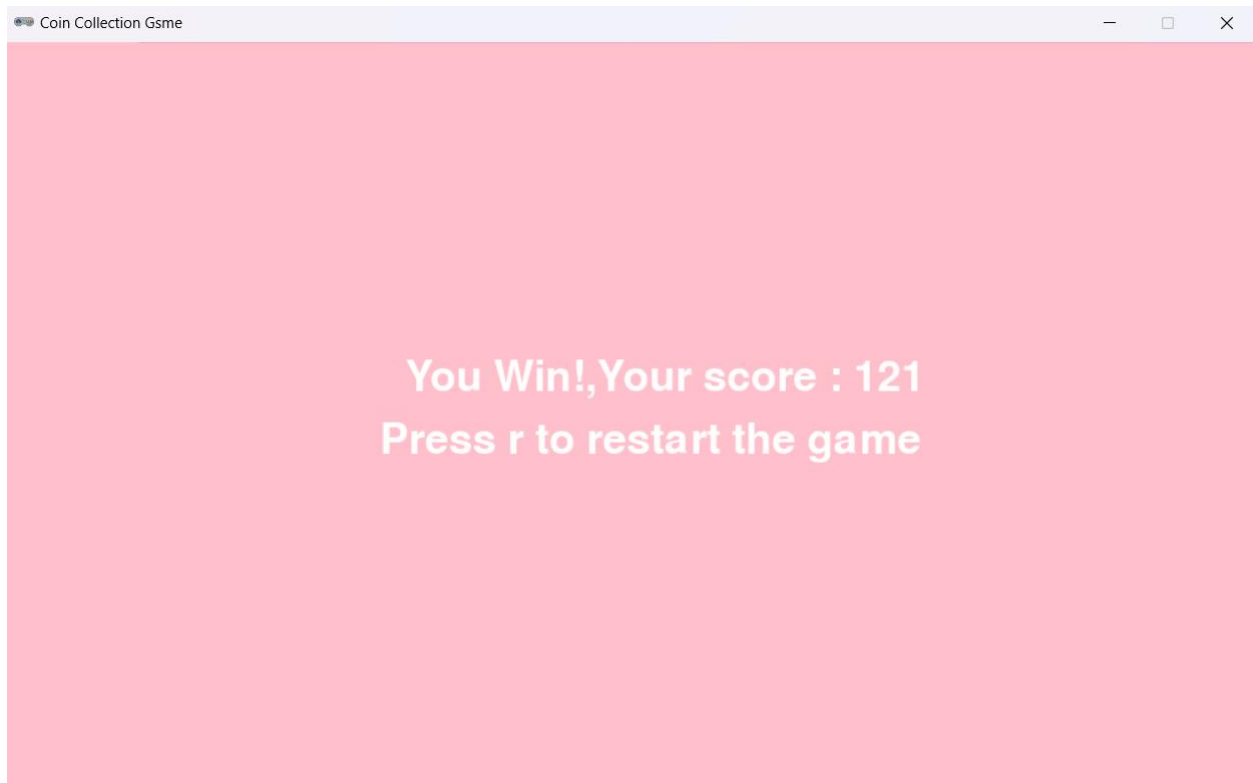
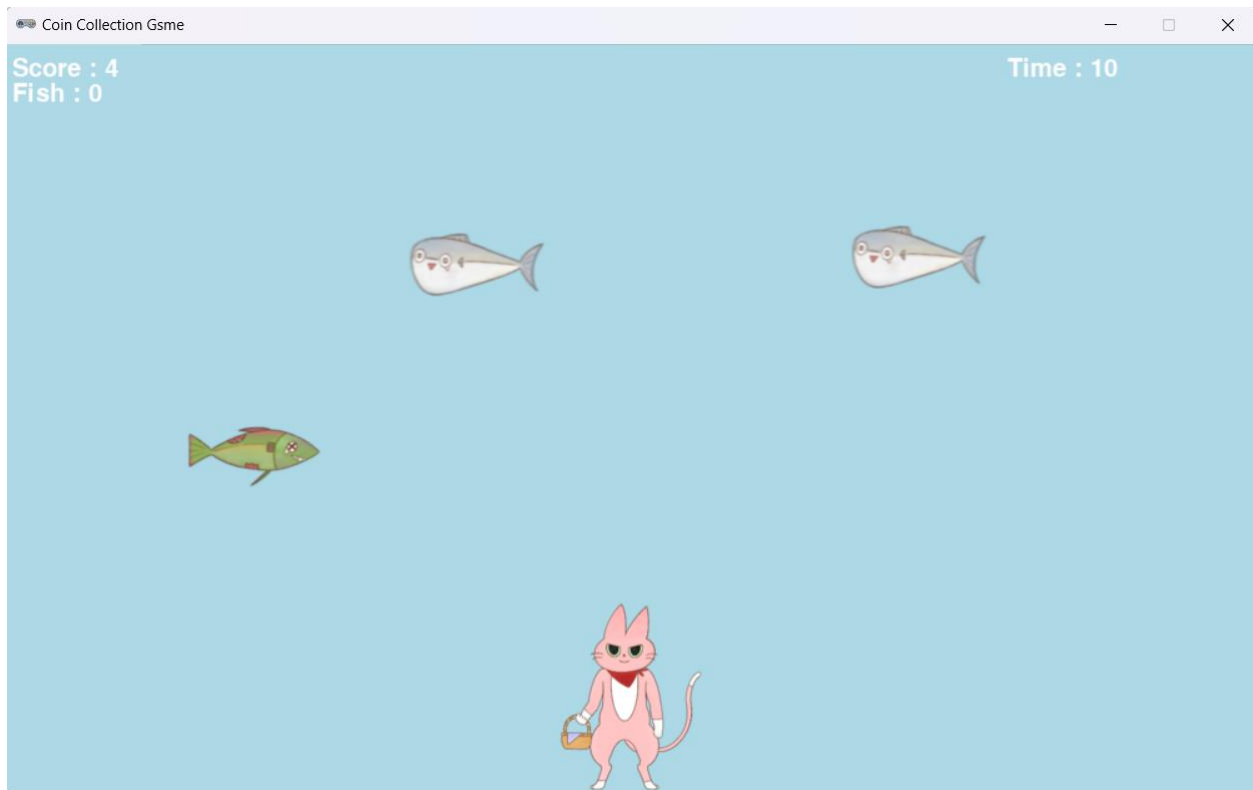
แนวคิด - เริ่มจากนำเกมอย่างง่ายในคาบเรียนมาพัฒนาต่อและเพิ่มฟังก์ชันและลูกเล่นเข้ามาทำให้เกมมีความน่าสนใจมากยิ่งขึ้น เกมนี้เป็นเกมแนว 2D ที่มีตัวแมวโดยจะให้ผู้เล่นรวบรวมเหรียญ และปลา เพื่อเพิ่มคะแนน - เกมมีคะแนน เวลา ค่าเงิน และจำนวนปลาเป็นตัวบ่งบอกสถานะเกม 2. การพัฒนา - การสร้างหน้าต่างเกม - โปรแกรมเริ่มต้นด้วยการสร้างหน้าต่างเกม โดยใช้ pgzrun เป็นเครื่องมือสำหรับการพัฒนาเกมบน Pygame Zero.

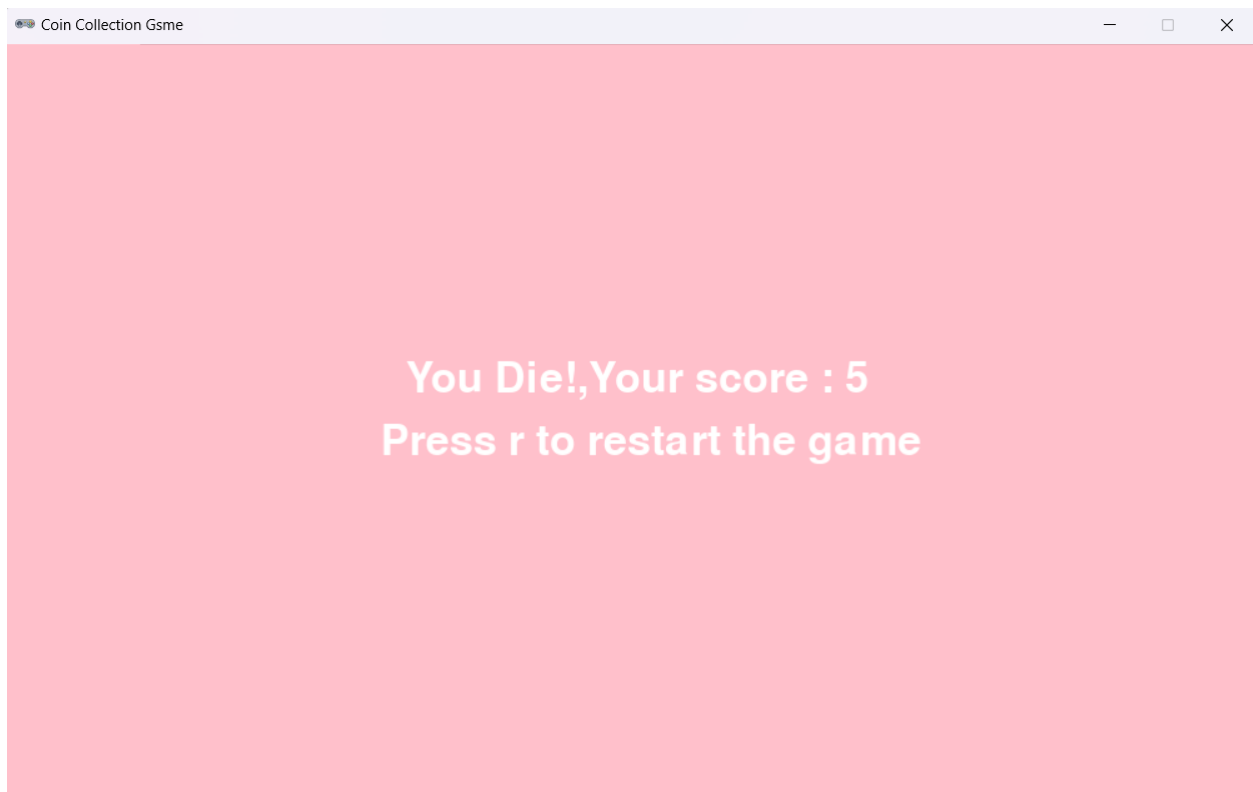
- การกำหนดค่าเริ่มต้น - กำหนดค่าเริ่มต้นของตัวแปรต่างๆ เช่น Score, Time, Fish, MAX_TIME และ Game_Over
- การวาดหน้าจอเกม (Draw Game Screen) - ในฟังก์ชัน draw() ทำการสร้างหน้าจอเกมโดยตรวจสอบสถานะ Game_Over - ถ้า Game_Over เป็น True แสดงหน้าจอสีชมพูพร้อมกับข้อความ "You Die!" หรือ "You Win!" และข้อความเพื่อเริ่มเกมใหม่ - ถ้า Game_Over เป็น False แสดงหน้าจอที่มีสีพื้นหลังต่างๆ และแสดงคะแนนจำนวนปลาและเวลา
- การอัปเดตสถานะเกม (Update Game State) - ในฟังก์ชัน update() ทำการอัปเดตสถานะของเกม เช่น ตัวแมวที่สามารถเคลื่อนที่ด้านซ้ายและขวา การเคลื่อนที่ของวัตถุเหรียญ และปลาลงมาจากด้านบนของหน้าจอ การตรวจสอบการชนของตัวแมวกับวัตถุเหรียญและปลา และอัปเดตคะแนน และจำนวนปลา
- การตรวจสอบเงื่อนไขสิ้นสุดเกม (Check for Game Over) - ในฟังก์ชัน update() ทำการตรวจสอบว่าเกมจบหรือไม่ โดยตรวจสอบการทำคะแนน เวลา และจำนวนปลา - หากเกมจบ จะกำหนด Game_Over เป็นข้อความ "You Die!" หรือ "You Win!" และหยุดการเรียกใช้ count_time()
- การรีเซ็ตเกม (Restart Game) - เมื่อผู้เล่นเลือกกรีตเกม โปรแกรมจะรีเซ็ตคะแนน เวลา และจำนวนปลา - โปรแกรมจะเริ่มเกมใหม่โดยเรียก count_time() และกำหนด Game_Over เป็น False
- สิ้นสุดเกม (Exit Game) - โปรแกรมสามารถออกจากเกมเมื่อผู้เล่นเลือกออกจากเกมจากหน้าจอสิ้นสุดเกม โดยรวม, โปรแกรมนี้เป็นตัวอย่างของการพัฒนาเกมง่ายๆ ด้วย Pygame Zero ที่ได้มีการพัฒนามาจากในคาบเรียนที่อาจารย์สอน ที่มีการจัดการการแสดงผล การอัปเดตสถานะเกม และ การตรวจสอบเงื่อนไขสิ้นสุดเกมเพื่อสร้างประสบการณ์เล่นเกมพื้นฐานใน Python

2.การออกแบบ Flow การทำงานโปรแกรม



3.หน้าเจอการทำงานของเกม





4.สรุปรายงานการทำงานของแต่ละบุคคลที่รับผิดชอบ

นางสาวณิชา ทองอยู่ รับผิดชอบการ เขียนโค้ด และ รายงาน

นางสาวกัญจนพร จตุรพรพรหม รับผิดชอบการ ออกแบบเกม เขียนโค้ดและรายงาน

นางสาววรินยุพา เข็ญมอยู่ รับผิดชอบการ การวาดรูป และการออกแบบของเกม

นางสาวเพ็ญภา คมะปะเต รับผิดชอบการ ออกแบบเกม และรายงาน

นางสาวพัชรภา ตุ่มเจริญ รับผิดชอบการ วาดรูปและออกแบบอนิเมชั่น ตัวละครในเกม และรายงาน

5. Source Program

```
import pgzrun
from random import randint

def draw():
    if Game_Over:
        screen.fill('pink')
        screen.draw.text(Game_Over,(320,250),fontsize=50)
        screen.draw.text('Press r to restart the
game',(300,300),fontsize=50)
    else:
        if Score <= 25:
            screen.fill(('light blue'))
            screen.draw.text(f'Score : {Score}',(5,10),fontsize=30)
            screen.draw.text(f'Fish : {Fish}',(5,30),fontsize=30)
            screen.draw.text(f'Time : {Time}',(800,10),fontsize=30)
            cat.draw()
            fd1.draw()
            fd2.draw()
            fn.draw()

        elif Score <= 50:
```

```
screen.fill((255,222,173))
screen.draw.text(f'Score : {Score}',(5,10),fontsize=30)
screen.draw.text(f'Fish : {Fish}',(5,30),fontsize=30)
screen.draw.text(f'Time : {Time}',(800,10),fontsize=30)
cat.draw()
fd1.draw()
fd2.draw()
fn.draw()
```

else:

```
screen.fill((230,230,250))
screen.draw.text(f'Score : {Score}',(5,10),fontsize=30)
screen.draw.text(f'Fish : {Fish}',(5,30),fontsize=30)
screen.draw.text(f'Time : {Time}',(800,10),fontsize=30)
cat.draw()
fd1.draw()
fd2.draw()
fn.draw()
```

def update():

```
    global Score,Fish,Game_Over
    if Game_Over: return;
```

```
if (keyboard.LEFT) : cat.x -= 20
elif (keyboard.RIGHT) : cat.x += 20
fd1.y += fd1.speed
if (fd1.y > HEIGHT):
    place_fd1()
fd2.y += fd2.speed
if (fd2.y > HEIGHT):
    place_fd2()
fn.y += fn.speed
if (fn.y > HEIGHT):
    place_fn()

if cat.x > WIDTH:
    cat.x = 0
if cat.x < 0:
    cat.x = WIDTH

if (cat.colliderect(fd1)):
    music.play_once('cion')
    Score += 1
    place_fd1()
```



```
if (cat.colliderect(fd2)):
```

```
    music.play_once('cion')
```

```
    Score += 1
```

```
    place_fd2()
```

```
if (cat.colliderect(fn)):
```

```
    print("collide")
```

```
    music.play_once('meoww')
```

```
    Score -= 2
```

```
    Fish += 1
```

```
    place_fn()
```

```
if Fish >= 5:
```

```
    Game_Over = f'You Die!,Your score : {Score}'
```

```
    clock.unschedule(count_time)
```

```
def on_key_down(key,mod,unicode):
```

```
    global Game_Over, Fish, Time, Score
```

```
    if Game_Over and key == keys.R:
```

```
        Score = 0
```

```
        Time = 0
```

```
        Fish = 0
```

```
        clock.schedule_interval(count_time,1.0)
```

```
Game_Over = False
```

```
def place_fd1():
```

```
    fd1.x = randint(fd1.width, WIDTH - fd1.width)
```

```
    fd1.y = 0
```

```
    fd1.speed = randint(8,10)
```

```
def place_fd2():
```

```
    fd2.x = randint(fd2.width, WIDTH - fd2.width)
```

```
    fd2.y = 0
```

```
    fd2.speed = randint(8,10)
```

```
def place_fn():
```

```
    if Score <= 25:
```

```
        fn.x = randint(0, WIDTH)
```

```
        fn.y = 0
```

```
        fn.speed = 7
```

```
    elif Score <= 50:
```

```
        fn.x = randint(0, WIDTH)
```

```
        fn.y = 0
```

```
        fn.speed = 10
```

else:

fn.x = randint(0, WIDTH)

fn.y = 0

fn.speed = 12

def count_time():

global Time,Game_Over

Time += 1

if Time == MAX_TIME:

Game_Over = f'You Win!,Your score : {Score}'

clock.unschedule(count_time)

#MAIN

TITLE = 'Coin Collection Gsme'

WIDTH,HEIGHT = 1000,600

cat = Actor('cat',(WIDTH/2,520))

Score = 0

Time = 0

Fish = 0

MAX_TIME = 80

Game_Over = False

fd1 = Actor('fd')

```
fd2 = Actor('fd')
```

```
fn = Actor('fn')
```

```
place_fd1()
```

```
place_fd2()
```

```
place_fn()
```

```
count_time()
```

```
clock.schedule_interval(count_time,1.0)
```

```
pgzrun.go()
```

Point of Sale : POS

1.แนวคิดและหลักการเกี่ยวกับการพัฒนาโปรแกรม

โปรแกรมที่เป็นตัวอย่างของระบบจุดขาย (Point of Sale) ที่ใช้ในการบริหารจัดการสินค้าและการขายสินค้า โปรแกรมนี้ใช้ภาษา Python เป็นฐานในการพัฒนา นี่คือนิยามและหลักการสำคัญของโปรแกรม

Class และ Object-Oriented Programming (OOP): - คุณสร้างคลาส **Product** เพื่อแทนสินค้าแต่ละชนิด โดยมีคุณสมบัติเช่น ชื่อสินค้า (name), ประเภท (category), ราคา (price), และสต็อก (stock) ซึ่งเป็นตัวแทนของแต่ละสินค้า - คุณสร้างคลาส **PointOfSale** เพื่อจัดการกับรายการสินค้า ซึ่งมีเมธอดเพิ่มสินค้า (**add_product**), แสดงรายการสินค้า (**list_products**), และขายสินค้า (**sell_product**)

การทำงานกับข้อมูล: - โปรแกรมให้ผู้ใช้เลือกทำการเพิ่มสินค้า, ดูรายการสินค้า, หรือขายสินค้าตามที่ต้องการ โดยใช้โครงสร้างควบคุมการทำงานแบบลูป (**while True**) เพื่อให้ผู้ใช้สามารถทำการเลือกต่อไปหรือออกจากโปรแกรมได้

การจัดการข้อผิดพลาด: - โปรแกรมมีการตรวจสอบว่าสินค้าที่ถูกสั่งซื้อมีจำนวนในสต็อกเพียงพอหรือไม่ และแจ้งเตือนให้ลูกค้าทราบถ้าสินค้าไม่เพียงพอ

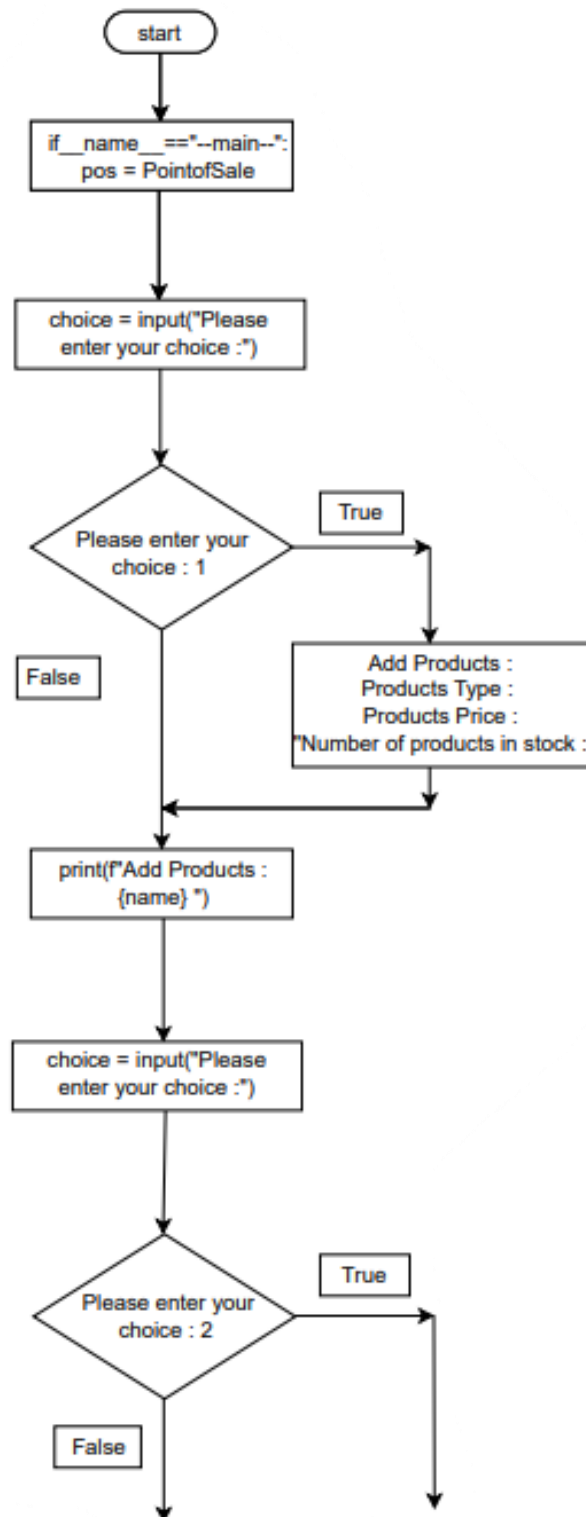
การแสดงผล: - โปรแกรมใช้ฟังก์ชัน **print** เพื่อแสดงข้อมูลและความในหน้าจอคอมพิวเตอร์

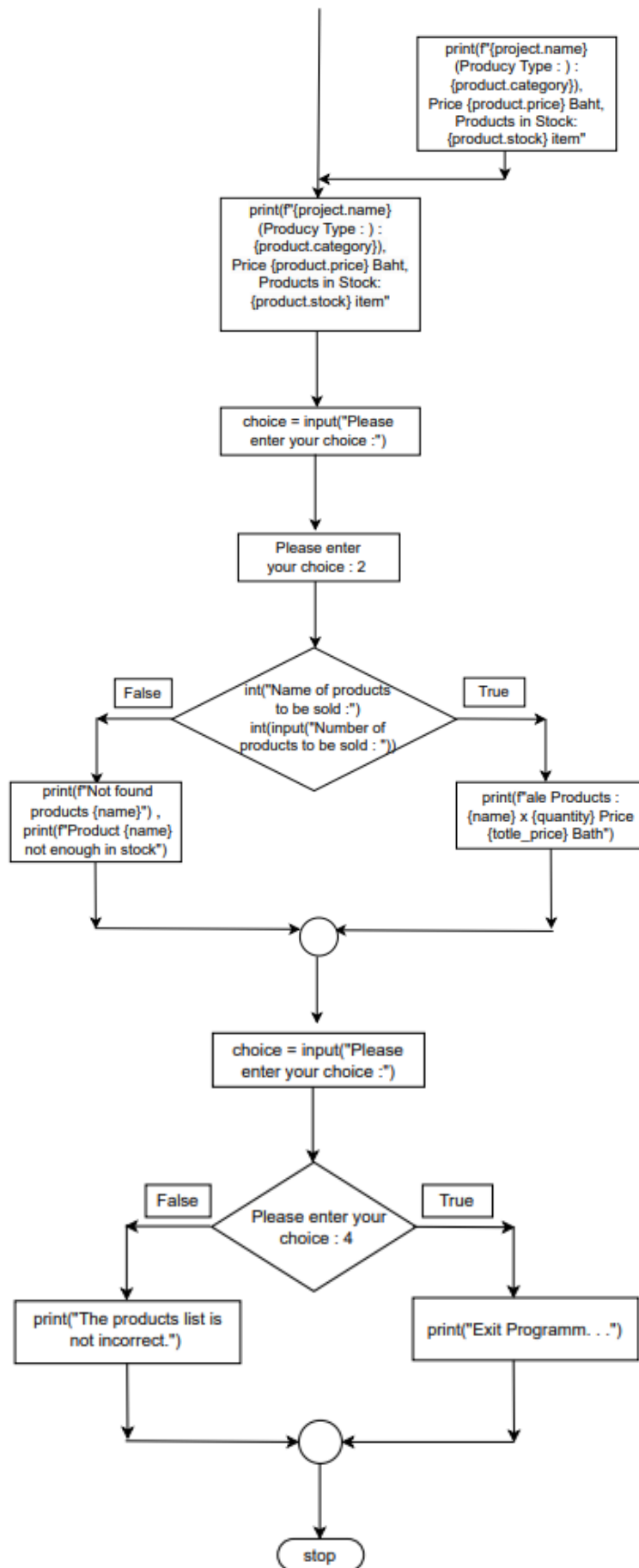
การใช้เงื่อนไขและการค้นหาข้อมูล: - โปรแกรมใช้โครงสร้างเงื่อนไข **if-elif-else** เพื่อตรวจสอบตัวเลือกที่ผู้ใช้เลือกและดำเนินการตามต้องการ - ในเมธอด **sell_product** โปรแกรมทำการค้นหาสินค้าที่ต้องการขายในรายการสินค้า

การนำเข้าและใช้งาน Class: - คุณสร้างวัตถุของคลาส **PointOfSale** และใช้เมธอดที่ถูกกำหนดไว้ในคลาส **PointOfSale** เพื่อจัดการกับข้อมูลสินค้า

ในสรุป, โปรแกรมที่คุณพัฒนาเป็นตัวอย่างของการใช้ Object-Oriented Programming (OOP) เพื่อจัดการข้อมูลและการทำงานกับข้อมูลในรูปแบบของระบบจุดขาย. โปรแกรมมีการจัดการกับข้อมูลสินค้า, การตรวจสอบจำนวนสินค้าในสต็อก, และการนำเสนอข้อมูลให้ผู้ใช้ในรูปแบบที่เข้าใจง่าย.

2.การออกแบบ Flow การทำงานโปรแกรม





3.สรุปรายงานการทำงานของแต่ละบุคคลที่รับผิดชอบ

นางสาวณิชา ทองอยู่ รับผิดชอบการ เขียนรายงาน

นางสาวกัญจนพร จตุรพรพรหม รับผิดชอบการ สรุปและรวบรวมข้อมูลของโค้ดและรายงาน

นางสาววิริยพา เจริญอยู่ รับผิดชอบการ การเขียนโค้ด และรายงาน

นางสาวเพ็ญภา คมมะปะเต รับผิดชอบการ การเขียนโค้ด และรายงาน

นางสาวพัชรภา ตุ่มเจริญ รับผิดชอบการ ออกแบบโปรแกรม

4. Source Program

class Product:

```
def __init__(self, name, category, price, stock):
```

```
    self.name = name
```

```
    self.category = category
```

```
    self.price = price
```

```
    self.stock = stock
```

class PointOfSale:

```
def __init__(self):
```

```
    self.products = []
```

```
def add_product(self, name, category, price, stock):
```

```
    product = Product(name, category, price, stock)
```

```
    self.products.append(product)
```

```
    print(f"Add Products : {name}")
```

```
def list_products(self):
```

```
    print("Products List :")
```



```
    for product in self.products:
        print(f"{product.name} (Product Type : {product.category}), Price {product.price} Baht, Products in Stock: {product.stock} item")
```

```
def sell_product(self, name, quantity):
    for product in self.products:
        if product.name == name:
            if product.stock >= quantity:
                product.stock -= quantity
                total_price = product.price * quantity
                print(f"Sale Products : {name} x {quantity} Price {total_price} Baht")
            else:
                print(f"Product {name} not enough in stock")
        return
    print(f"Not found products {name}")
```

```
if __name__ == "__main__":
    pos = PointOfSale()
    while True:
        print("\n")
        print("="*22)
        print(":   Main Menu   :")
        print("="*22)
        print(": 1. Add Products   :")
        print(": 2. Products List  :")
        print(": 3. Sale Products  :")
        print(": 4. Exit Program   :")
        print("="*22)
```

```
choice = input("Please enter your choice : ")

if choice == "1":
    name = input("Add Products : ")
    category = input("Products Type : ")
    price = float(input("Products Price : "))
    stock = int(input("Number of products in stock : "))
    pos.add_product(name, category, price, stock)
elif choice == "2":
    pos.list_products()
elif choice == "3":
    name = input("Name of products to be sold : ")
    quantity = int(input("Number of products to be sold : "))
    pos.sell_product(name, quantity)
elif choice == "4":
    print("Exit Program. . .")
    break
else:
    print("The products list is not incorrect.")
```