

HANGMAN GAME PROJECT

Computer Programming Languages (204217)





MEMBERS

1. **สุธิมา เฟื่องพินิจ** **590510137**
2. **วรรณรัตน์ บุณยวงศ์** **600510125**
3. **เจษฎา อิกทาสน** **600510538**

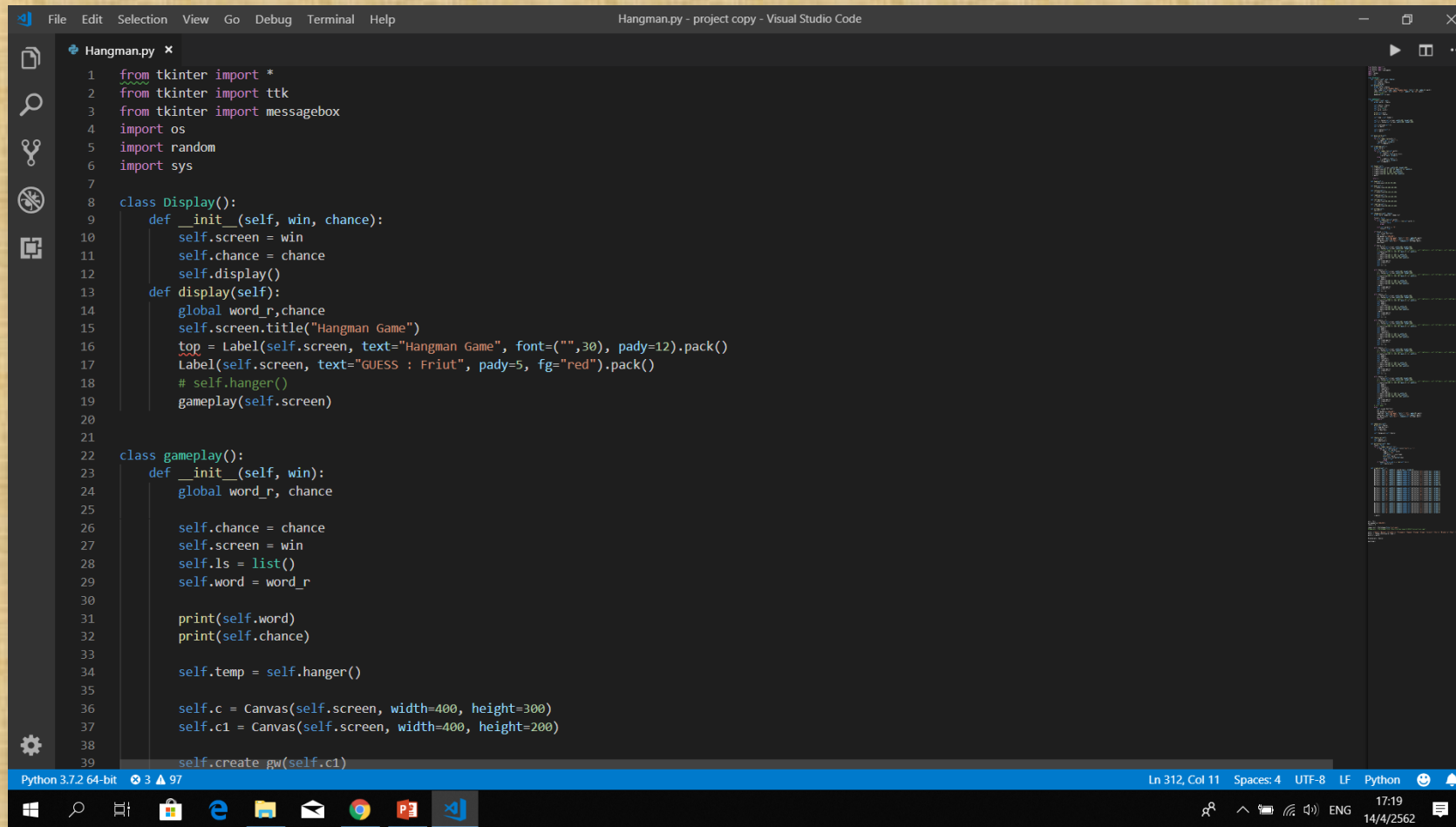
Inspiration



กลุ่มของพวกเราได้สนใจที่จะสร้างเกม Hangman ขึ้นมา
เนื่องจาก ความยากในการสร้างอยู่ในระดับปานกลาง
ไม่ยากหรือไม่ง่ายจนเกินไป เป็นเกมที่เป็นที่รู้จักของทุกคนทั่วไป และสามารถฝึกทักษะในด้าน
คำศัพท์ภาษาอังกฤษได้



Source Code



```
1 from tkinter import *
2 from tkinter import ttk
3 from tkinter import messagebox
4 import os
5 import random
6 import sys
7
8 class Display():
9     def __init__(self, win, chance):
10         self.screen = win
11         self.chance = chance
12         self.display()
13     def display(self):
14         global word_r, chance
15         self.screen.title("Hangman Game")
16         top = Label(self.screen, text="Hangman Game", font=("",30), pady=12).pack()
17         Label(self.screen, text="GUESS : Friut", pady=5, fg="red").pack()
18         # self.hanger()
19         gameplay(self.screen)
20
21
22 class gameplay():
23     def __init__(self, win):
24         global word_r, chance
25
26         self.chance = chance
27         self.screen = win
28         self.ls = list()
29         self.word = word_r
30
31         print(self.word)
32         print(self.chance)
33
34         self.temp = self.hanger()
35
36         self.c = Canvas(self.screen, width=400, height=300)
37         self.c1 = Canvas(self.screen, width=400, height=200)
38
39         self.create_gw(self.c1)
```



Classify

- class Display

หน้าตาแสดงเกม Hangman ขึ้นมา

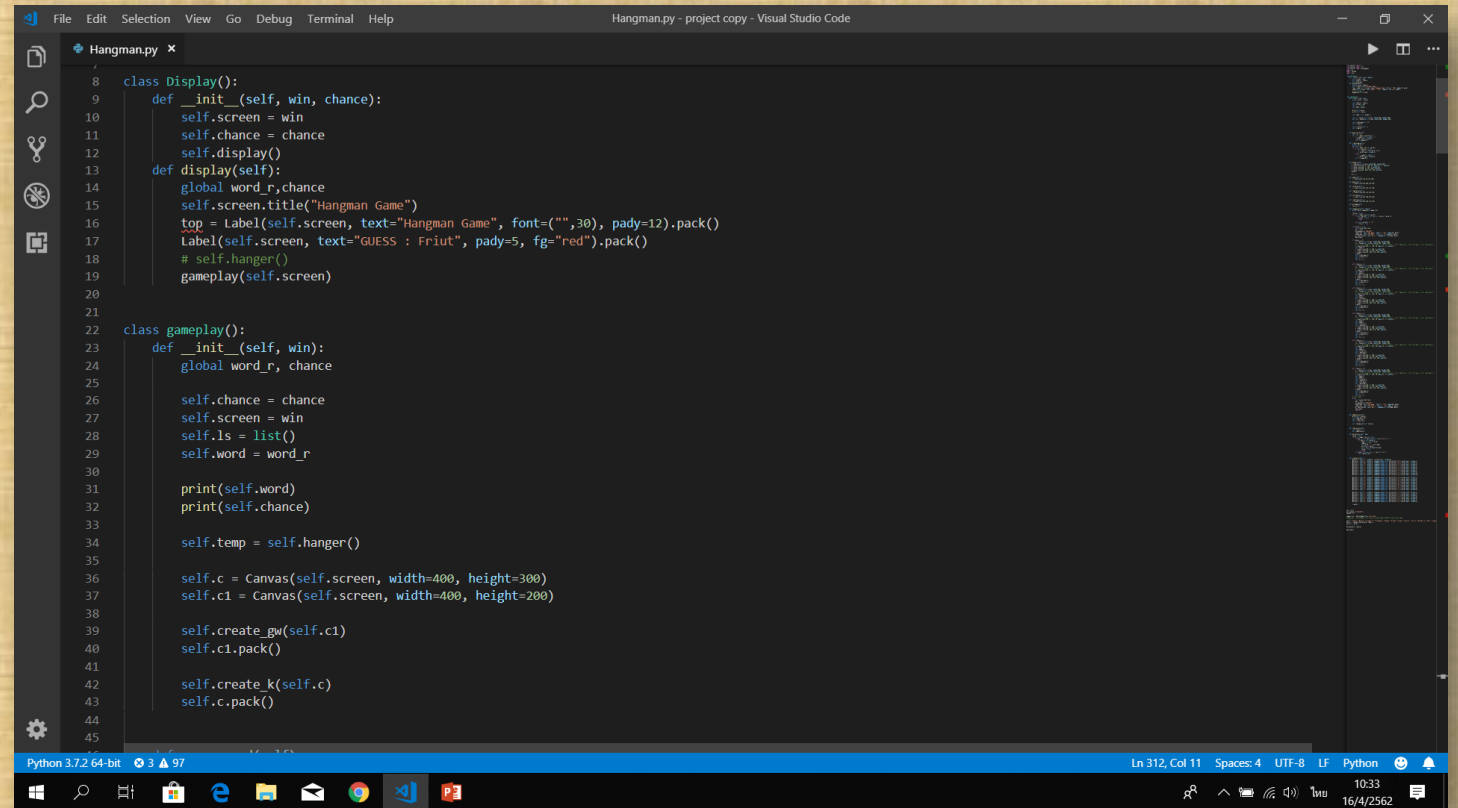
ทั้งปุ่มกด การวาดภาพ

- class gameplay

เป็นclassที่มีฟังก์ชันต่าง ๆ รวมอยู่

ในนี้ในการวาดรูป ดำศัพท์ กฎเกณฑ์

ในการแพ้-ชนะ



```
8 class Display():
9     def __init__(self, win, chance):
10         self.screen = win
11         self.chance = chance
12         self.display()
13     def display(self):
14         global word_r, chance
15         self.screen.title("Hangman Game")
16         top = Label(self.screen, text="Hangman Game", font=("", 30), pady=12).pack()
17         Label(self.screen, text="GUESS : Friut", pady=5, fg="red").pack()
18         # self.hanger()
19         gameplay(self.screen)
20
21
22 class gameplay():
23     def __init__(self, win):
24         global word_r, chance
25
26         self.chance = chance
27         self.screen = win
28         self.ls = list()
29         self.word = word_r
30
31         print(self.word)
32         print(self.chance)
33
34         self.temp = self.hanger()
35
36         self.c = Canvas(self.screen, width=400, height=300)
37         self.c1 = Canvas(self.screen, width=400, height=200)
38
39         self.create_gw(self.c1)
40         self.c1.pack()
41
42         self.create_k(self.c)
43         self.c.pack()
44
45
```



Function

ยกตัวอย่างฟังก์ชัน

- ฟังก์ชัน create_k คือการสร้างปุ่ม a-z

ใช้แสดงภาพปุ่มในหน้าต่างเพื่อทดสอบตัวอักษร

ในารมย

```
File Edit Selection View Go Debug Terminal Help Hangman.py - project copy - Visual Studio Code

def create_k(self, c):
    Button(c, text="q", width=2, command=lambda:self.buttonfunc("q")).grid(row=1, column=1)
    Button(c, text="w", width=2, command=lambda:self.buttonfunc("w")).grid(row=1, column=2)
    Button(c, text="e", width=2, command=lambda:self.buttonfunc("e")).grid(row=1, column=3)
    Button(c, text="r", width=2, command=lambda:self.buttonfunc("r")).grid(row=1, column=4)
    Button(c, text="t", width=2, command=lambda:self.buttonfunc("t")).grid(row=1, column=5)
    Button(c, text="y", width=2, command=lambda:self.buttonfunc("y")).grid(row=1, column=6)
    Button(c, text="u", width=2, command=lambda:self.buttonfunc("u")).grid(row=1, column=7)
    Button(c, text="l", width=2, command=lambda:self.buttonfunc("l")).grid(row=1, column=8)
    Button(c, text="o", width=2, command=lambda:self.buttonfunc("o")).grid(row=1, column=9)
    Button(c, text="p", width=2, command=lambda:self.buttonfunc("p")).grid(row=1, column=10)

    Button(c, text="a", width=2, command=lambda:self.buttonfunc("a")).grid(row=2, column=1)
    Button(c, text="s", width=2, command=lambda:self.buttonfunc("s")).grid(row=2, column=2)
    Button(c, text="d", width=2, command=lambda:self.buttonfunc("d")).grid(row=2, column=3)
    Button(c, text="f", width=2, command=lambda:self.buttonfunc("f")).grid(row=2, column=4)
    Button(c, text="g", width=2, command=lambda:self.buttonfunc("g")).grid(row=2, column=5)
    Button(c, text="h", width=2, command=lambda:self.buttonfunc("h")).grid(row=2, column=6)
    Button(c, text="j", width=2, command=lambda:self.buttonfunc("j")).grid(row=2, column=7)
    Button(c, text="k", width=2, command=lambda:self.buttonfunc("k")).grid(row=2, column=8)
    Button(c, text="i", width=2, command=lambda:self.buttonfunc("i")).grid(row=2, column=9)

    Button(c, text="z", width=2, command=lambda:self.buttonfunc("z")).grid(row=3, column=2)
    Button(c, text="x", width=2, command=lambda:self.buttonfunc("x")).grid(row=3, column=3)
    Button(c, text="c", width=2, command=lambda:self.buttonfunc("c")).grid(row=3, column=4)
    Button(c, text="v", width=2, command=lambda:self.buttonfunc("v")).grid(row=3, column=5)
    Button(c, text="b", width=2, command=lambda:self.buttonfunc("b")).grid(row=3, column=6)
    Button(c, text="n", width=2, command=lambda:self.buttonfunc("n")).grid(row=3, column=7)
    Button(c, text="m", width=2, command=lambda:self.buttonfunc("m")).grid(row=3, column=8)

    c.pack()
```

- ฟังก์ชัน change_pic คือการสร้างภาพ Hangman

เมื่อมีการคลิกตัวอักษรผิดก็จะเริ่มห้อยหัว ฟังก์ชันจะ

ทำงานโดยการโยนจำนวนครั้งที่ผิดเข้าไปและแสดงผลภาพ

ออกมา

```
File Edit Selection View Go Debug Terminal Help Hangman.py - project copy - Visual Studio Code

def change_pic(self, chance):
    global word, image_win, image_lost

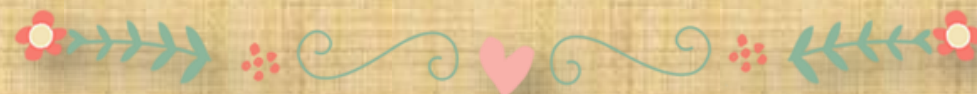
    finish = False
    for i in range(len(self.word)):
        if self.word[i] != "#" and i < (len(self.word)-1):
            finish = False
            break

        elif self.word[i] == "#":
            finish = True

    if finish == True:
        self.screen.destroy()
        win = Tk()
        win.geometry("200x200")
        label(win, text="End game", font=("",45"), pady=30).pack()
        label(win, text="YOU WIN!", font=("",18"),pady=5).pack()
        q = Button(win, text="Quit!", command=self.exitApp).pack()
        mainloop()

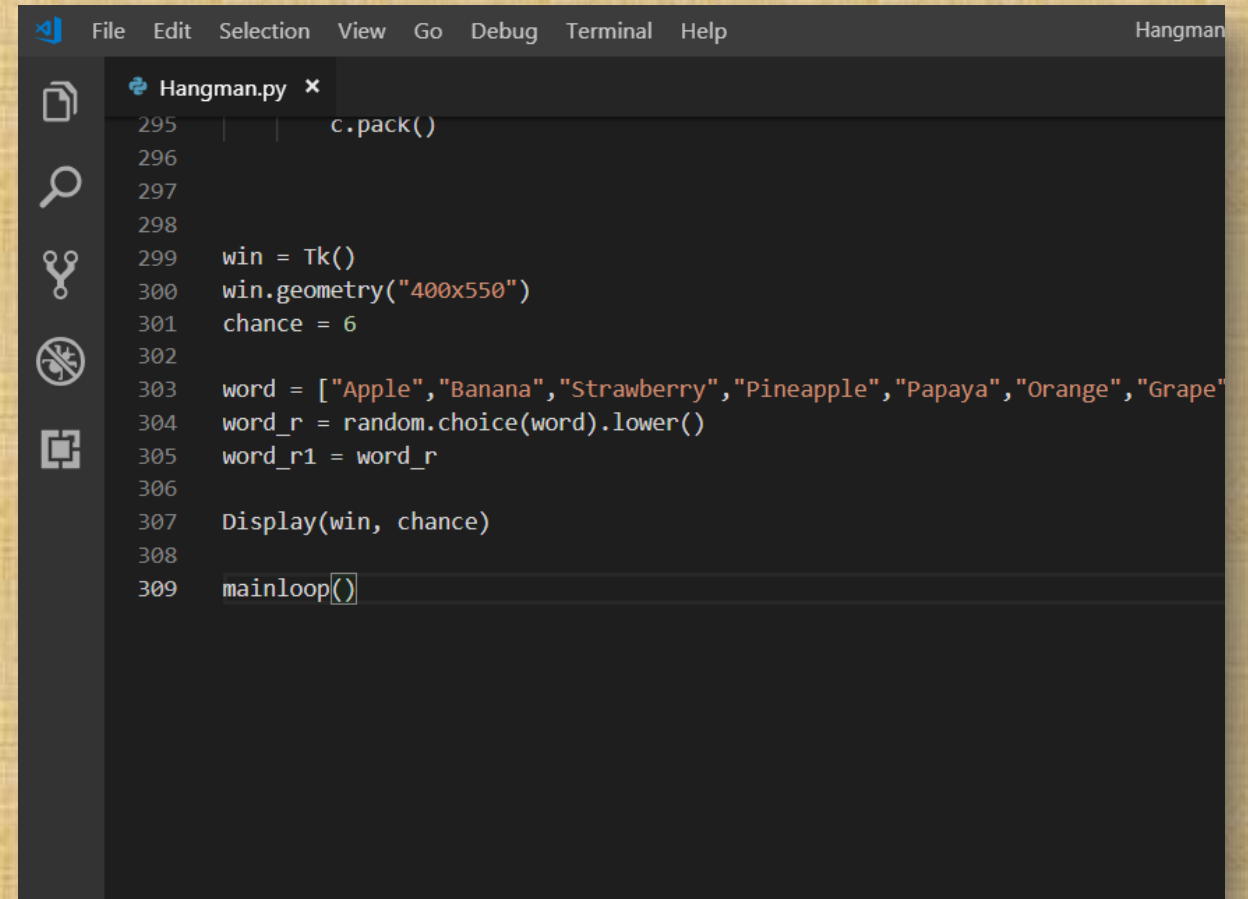
    if chance == 5:
        c = Canvas(self.screen, width=400, height=300)
        c1 = Canvas(self.screen, width=400, height=200)
        # ls = [self.head(c), self.body(c), self.leftarm(c), self.rightarm(c), self.leftleg(c), self.rightleg(c)]
        c.create_line(140, 3, 140, 40, dash=(4, 2), width=2)
        self.head(c)
        c.create_line(140, 3, 250, 3, width=10)
        c.create_line(250, 0, 250, 280, width=6)
        c.create_line(130, 280, 310, 280, width=6)
        c.pack()

        self.create_gw(c1)
        self.create_k(c1)
        self.c = c
        self.c1 = c1
```



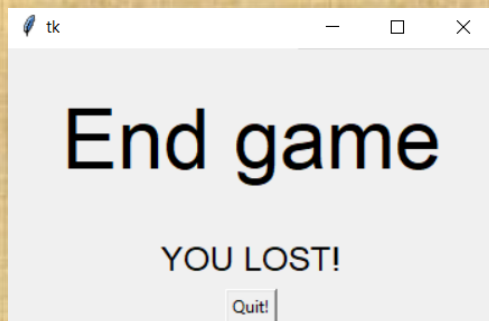
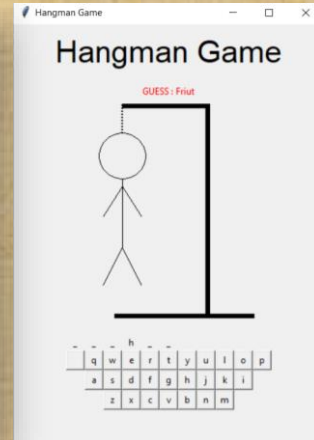
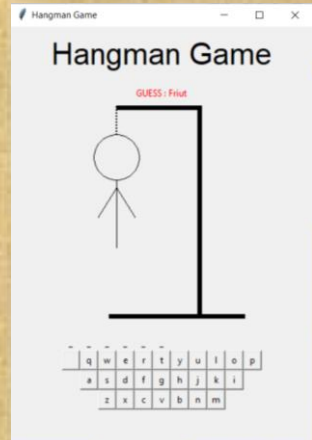
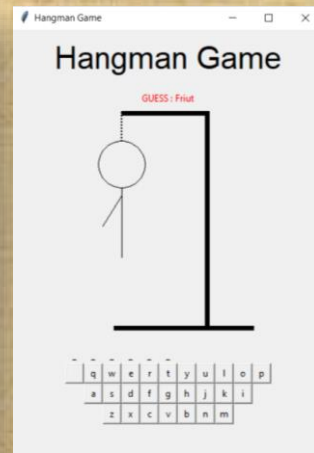
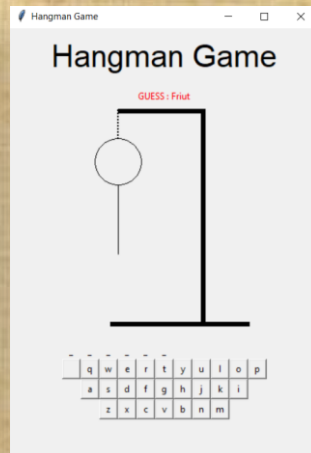
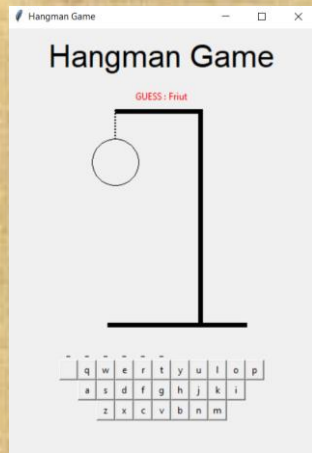
Main

Win คือ ตัวแปรเพื่อเปิด module Tkinter ที่เราได้ import เข้ามา สร้างหน้าต่างขนาด 400x500 ส่วน chance คือ ค่าที่ตั้งไว้สำหรับการนับจำนวนที่ผิด word คือ ตัวแปรที่เราใช้เก็บ list ของคำศัพท์ word_r คือ ตัวแปรรับคำศัพท์ที่ random แล้วเปิด class Display โดยโยน win และ chance เข้าไปจากนั้นก็ทำตามขั้นตอน function ต่างๆ ใน class



```
295 c.pack()
296
297
298
299 win = Tk()
300 win.geometry("400x550")
301 chance = 6
302
303 word = ["Apple", "Banana", "Strawberry", "Pineapple", "Papaya", "Orange", "Grape"]
304 word_r = random.choice(word).lower()
305 word_r1 = word_r
306
307 Display(win, chance)
308
309 mainloop()
```





HOW TO PLAY



เมื่อเกม Start เกมจะทำการสุ่มคำศัพท์มา 1 คำ

ให้ผู้เล่นเดาตัวอักษรภาษาอังกฤษมา 1 ตัว

ถ้าเดาถูกต้องตัวอักษรนั้นจะเป็น 1 ในตัวที่อยู่ในคำศัพท์นั้นๆ ถ้าเดาผิด จะเริ่มแขวนคอไปเรื่อยๆ จนพิดเกิน 6 ครั้ง เกมจะจบทันที

Hint : คำศัพท์หมวดผลไม้

