

TRƯỜNG ĐẠI HỌC BẠC LIÊU
KHOA KỸ THUẬT VÀ CÔNG NGHỆ



ĐỒ ÁN 1

TÊN ĐỀ TÀI

CHƯƠNG TRÌNH TRÒ CHƠI ĐOÁN MÀU

Sinh viên thực hiện

Nguyễn Trường Khang

Mã sinh viên

227480201058

Giảng viên hướng dẫn

Ths. Trần Phước Nghĩa

HỌC KỲ I, 2024 - 2025

TRƯỜNG ĐẠI HỌC BẠC LIÊU
KHOA KỸ THUẬT VÀ CÔNG NGHỆ



ĐỒ ÁN 1

TÊN ĐỀ TÀI

CHƯƠNG TRÌNH TRÒ CHƠI ĐOÁN MÀU

Sinh viên thực hiện

Nguyễn Trường Khang

Mã sinh viên

227480201058

Giảng viên hướng dẫn

Ths. Trần Phước Nghĩa

HỌC KỲ I, 2024 - 2025

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

GIÁO VIÊN NHẬN XÉT

ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN ĐỒ ÁN 1

(Học kỳ I, Năm học 2024 - 2025)

TÊN ĐỀ TÀI: CHƯƠNG TRÌNH TRÒ CHƠI ĐOÁN MÀU

GIÁO VIÊN HƯỚNG DẪN:

STT	HỌ VÀ TÊN	MSCB
1	Ths. Trần Phước Nghĩa	

SINH VIÊN THỰC HIỆN:

STT	HỌ VÀ TÊN	MSSV	THƯỜNG (Tối đa 1,0 điểm)	ĐIỂM
1	Nguyễn Trường Khang	227480201058		

I. HÌNH THỨC (Tối đa 0,5 điểm)	
Bìa (tối đa 0,25 điểm)	
Các tiêu đề: Trường ĐHBL, Khoa CNTT Loại đồ án: 1 Tên đề tài Thông tin về các sinh viên thực hiện: họ tên, mã số sinh viên Giáo viên hướng dẫn: chức danh, họ tên Học kỳ, năm học thực hiện	
Bố cục (tối đa 0,25 điểm)	
Nhận xét, đánh giá của giảng viên chấm Mục lục: cấu trúc chương, mục và tiểu mục Phụ lục (nếu có) Tài liệu tham khảo	
II. NỘI DUNG (Tối đa 3,5 điểm)	
Tổng quan (tối đa 0,5 điểm)	
Mô tả bài toán, mục tiêu cần đạt được (0,25 điểm) Hướng giải quyết và kế hoạch thực hiện (0,25 điểm)	
Lý thuyết (tối đa 0,5 điểm)	

Các khái niệm sử dụng trong đề tài (0,25 điểm)	
Kết quả vận dụng lý thuyết vào đề tài (0,25 điểm)	
Ứng dụng (tối đa 2,0 điểm)	
Phân tích yêu cầu bài toán, xây dựng các cấu trúc dữ liệu cần thiết (tối đa 0,5 điểm)	
Giải thuật (Lưu đồ - Ngôn ngữ giả) (1,0 điểm)	
Giới thiệu chương trình (0,5 điểm)	
Kết luận (tối đa 0,5 điểm)	
Nhận xét kết quả đạt được	
Hạn chế	
Hướng phát triển	
III. CHƯƠNG TRÌNH DEMO (Tối đa 5,0 điểm)	
Giao diện thân thiện với người dùng (1,0 điểm)	
Hướng dẫn sử dụng (0,5 điểm)	
Kết quả thực hiện đúng với kết quả của phần ứng dụng (3,5 điểm)	

Ghi chú:

- Điểm trong khung “các sinh viên thực hiện” là điểm kết quả cuối cùng của từng sinh viên trong quá trình thực hiện đồ án

- Nếu sinh viên demo chương trình và trả lời vấn đáp không đạt yêu cầu của giáo viên hướng dẫn thì sinh viên sẽ nhận điểm F cho học phần này.

Bạc Liêu, ngày tháng năm 2024

GIẢNG VIÊN CHẤM

MỤC LỤC

CHƯƠNG 1 :TỔNG QUAN	1
1.1. Đặc tả đề tài	1
1.2. Yêu cầu của đề tài	1
1.3. Hướng giải quyết	1
1.4. Kế hoạch thực hiện	1
CHƯƠNG 2 : LÝ THUYẾT	2
2.1. Các khái niệm sử dụng trong đề tài	2
2.1.1. Mục tiêu cần đạt	2
2.1.2. Lý thuyết về trò chơi	2
2.1.3. Xây dựng mảng chứa màu ngẫu nhiên (không trùng màu)	2
2.2. Kết quả vận dụng lý thuyết vào đề tài	3
2.2.1. Khai báo thư viện	3
2.2.2. Khởi tạo Pygame và Pygame mixer	3
2.2.3. Tải âm thanh	3
2.2.4. Khai báo các biến và hằng số	4
2.2.5. Khởi tạo cửa sổ	4
2.2.6. Font chữ	4
2.2.7. Các mục trong menu	5
2.2.8. Hàm vẽ trò chơi	5
2.2.9. Hàm hiển thị game Over	5
2.2.10. Hàm vòng lặp trò chơi (Hàm quan trọng)	5
2.2.11. Hàm hiển thị giới thiệu	7
2.2.12. Hàm hướng dẫn	8
2.2.13. Hàm vẽ menu chính	8
2.2.14. Hàm quản lý menu	8
2.2.15. Khởi chạy chương trình	8
2.2.16. Thực thi demo	8
Chương 3: ỨNG DỤNG	9
3.1. Phân tích yêu cầu bài toán và xây dựng các cấu trúc dữ liệu	9
3.1.1. Phân tích yêu cầu	9
3.1.2. Cấu trúc dữ liệu	9
3.2. Thiết kế giải thuật (Lưu đồ - Ngôn ngữ giả)	10
3.2.1. Lưu đồ và ngôn ngữ giả chương trình chính	10
3.2.2. Lưu đồ và sơ đồ giả chương trình tạo màu ngẫu nhiên	11
3.2.3. Lưu đồ và sơ đồ giả chương trình đoán màu	13

3.3. Giới thiệu về chương trình	15
CHƯƠNG 4: KẾT LUẬN.....	22
4.1. Nhận xét kết quả đạt được.....	22
4.2. Những hạn chế.....	22
4.3. Hướng phát triển.....	22
PHỤ LỤC	23
TÀI LIỆU THAM KHẢO.....	26

PHẦN MỞ ĐẦU

Hiện nay, giáo dục đại học không chỉ dừng lại ở việc truyền đạt kiến thức mà chú trọng hơn vào việc rèn luyện tư duy cho sinh viên. Ở các cấp học thấp hơn như trung học phổ thông, các phương pháp giảng dạy truyền thống cũng dần được thay thế bằng cách tiếp cận mới, lấy người học làm trung tâm. Giáo viên đóng vai trò như người định hướng, khơi gợi khả năng tự học và sáng tạo.

Phương pháp này nhằm mục đích giúp học sinh, sinh viên phát triển tư duy – một yếu tố quan trọng trong việc giải quyết vấn đề, xử lý tình huống, và thích nghi với những thay đổi trong cuộc sống lẫn công việc. Một người có tư duy tốt sẽ luôn có lợi thế trong việc đưa ra quyết định và đối phó với các thách thức.

Có nhiều phương pháp để rèn luyện tư duy, từ việc tổ chức học nhóm để học sinh tự nghiên cứu, luyện tập khả năng tính nhẩm, đến rèn luyện trí nhớ qua việc hồi tưởng ký ức. Những hoạt động này không chỉ giúp nâng cao khả năng tư duy mà còn phát triển trí thông minh – điều đặc biệt quan trọng với học sinh, sinh viên. Để rèn luyện tư duy, một phương pháp thú vị và hấp dẫn là thông qua các trò chơi mang tính suy luận và logic cao. Một trong số đó là trò chơi **“Đoán màu”**. Trò chơi không chỉ giúp rèn luyện trí tuệ mà còn thách thức khả năng tư duy logic của người chơi.

Đặc tả đề tài

Trò chơi “Đoán màu” được thiết kế như sau:

Sử dụng từ 4 đến 7 màu sắc.

Người chơi có tối đa mười lượt đoán, mỗi lần đoán 4 màu.

Sau mỗi lần đoán, máy sẽ thông báo số vị trí đoán đúng.

Nếu sau mười lần đoán, người chơi không đoán đúng toàn bộ, kết quả sẽ được tính là thua.

Sau hơn hai tháng làm việc chăm chỉ, em đã hoàn thành đồ án này với những nội dung cơ bản đáp ứng yêu cầu đề tài. Thành công này có được là nhờ sự nỗ lực của bản thân, cùng với sự hướng dẫn tận tình từ Thầy.

Tuy nhiên, em hiểu rằng đồ án không thể tránh khỏi những thiếu sót do hạn chế về kiến thức và thời gian. Vì vậy, em rất mong nhận được những ý kiến đóng góp quý báu từ Thầy để có thể hoàn thiện hơn trong tương lai. Em xin gửi lời cảm ơn chân thành đến quý Thầy đã hỗ trợ em trong suốt quá trình thực hiện đồ án này.

Trân trọng,
Nguyễn Trường Khang

CHƯƠNG 1 :TỔNG QUAN

1.1. Đặc tả đề tài

Trò chơi “Đoán màu” được thể hiện : Có từ 4 đến 7 màu, người chơi một lần chỉ đoán 4 màu, có mười lần đoán. Máy tính sẽ đánh giá mỗi lần đoán và cho biết số vị trí đoán đúng. Sau mười lần đoán máy tính sẽ cho kết quả người chơi thắng hay thua.

1.2. Yêu cầu của đề tài

- Nắm vững cơ sở lý thuyết về cấu trúc dữ liệu. Các giải thuật sắp xếp.
- Chương trình cần có các chức năng như sau đối với các giải thuật:
 - Đọc dữ liệu từ file văn bản, xuất ra file văn bản
 - Cho biết số lần đổi chỗ của các giải thuật
 - So sánh các giải thuật với nhau

1.3. Hướng giải quyết

Về lý thuyết: nắm vững ngôn ngữ lập trình, xây dựng giải thuật đúng và phù hợp với yêu cầu đề tài.

Về chương trình: Sử dụng ngôn ngữ Python (Pycharm) cài đặt chương trình theo yêu cầu đề tài, và giao diện xây dựng trên màn hình đồ họa.

1.4. Kế hoạch thực hiện

Tìm hiểu về lý thuyết: 1 tuần

Xây dựng giải thuật: 2 tuần

Thiết kế giao diện: 1 tuần

Viết chương trình: 3 tuần

Viết báo cáo và hoàn chỉnh chương trình: 1 tuần

CHƯƠNG 2 : LÝ THUYẾT

2.1. Các khái niệm sử dụng trong đề tài

2.1.1. Mục tiêu cần đạt

Theo yêu cầu đề tài toát lên 3 vấn đề quan trọng sau: **Trước hết:** phải có một dãy 4 màu được tạo ra **một cách ngẫu nhiên và không trùng màu** từ một danh sách màu sẵn có cho người chơi đoán, **thứ hai:** chọn lựa màu từ phía người chơi, **thứ ba:** so sánh sự lựa chọn của người chơi và dãy ngẫu nhiên đó. **Ngoài ra** việc thiết kế giao diện, người dùng cũng là vấn đề quan trọng giúp trò chơi thành công hơn.

2.1.2. Lý thuyết về trò chơi

Theo yêu cầu của đề tài thì ta có về ví dụ về cách chơi như sau:

Tạo mảng chứa màu ngẫu nhiên (không trùng màu) và danh sách màu đoán từ người chơi.

M là số màu của trò chơi

N là số lần đoán của người chơi

- M từ 4 đến 7; tức là $\text{Min}(M) = 4$ và $\text{Max}(M) = 7$.

- Thành phần M màu được chọn: Đỏ, Xanh dương, Xanh lá, Vàng, Cam, Tím, Xanh da trời.

- N từ 4 đến 7; tức là $\text{Min}(N) = 4$ và $\text{Max}(N) = 7$

- **Các vòng chơi:**

Vòng 1 ứng với $M = 4$, gồm 4 màu: Đỏ, Xanh dương, Xanh lá, Vàng.

Vòng 2 ứng với $M = 5$, gồm 5 màu: Đỏ, Xanh dương, Xanh lá, Vàng, Cam.

Vòng 3 ứng với $M = 6$, gồm 6 màu: Đỏ, Xanh dương, Xanh lá, Vàng, Cam, Tím.

Vòng 4 ứng với $M = 7$, gồm 7 màu: Đỏ, Xanh dương, Xanh lá, Vàng, Cam, Tím, Xanh da trời.

=> Như vậy có 4 vòng chơi, N và M sẽ +1 theo mỗi vòng để độ khó được tăng lên.

2.1.3. Xây dựng mảng chứa màu ngẫu nhiên (không trùng màu)

Ý đồ tạo ra một dãy màu ngẫu nhiên không trùng nhau, cụ thể là 4 (+1 theo level) màu được chọn từ một dãy màu cho trước.

Trong các ngôn ngữ lập trình như C, Pascal, VS có một hàm được định nghĩa trước thỏa mãn tính chất trên – đó là hàm ngẫu nhiên. Hàm ngẫu nhiên trong Python là **import random** và nếu muốn không trùng với nhau thì hàm sẽ có cú pháp như sau:

`random.sample(population, k)`: Chọn k phần tử ngẫu nhiên từ một dãy mà không lặp lại.

VD:

```
items = [1, 2, 3, 4, 5]
print(random.sample(items, 3)) # Ví dụ: [4, 1, 3]
```

Nhận thấy đây sẽ là sự chọn lựa tốt nhất cho việc xây dãy màu ngẫu nhiên. Với ý tưởng như sau:

Mỗi lần dùng hàm sẽ được một con số, con số đó như một từ khóa cho ta một màu nào đó tự ta định nghĩa. Thí dụ: kết quả là 0 sẽ là màu đỏ, 1 sẽ cho ta màu xanh,... và ta chỉ việc lưu lại kết quả. Khi cho vòng lặp chạy 4 lần ta sẽ có 4 màu (cứ +1 khi lên level). Ta dùng tham số thực tế trong hàm Random chính là số màu của từng vòng chơi. Cụ thể viết tổng quát Random(<Số màu>); Số màu là tổng số màu của dãy mà máy phải chọn, nói cách khác nó là tổng số màu của một vòng nào đó. Có thể dùng một mảng để lưu kết quả.

2.2. Kết quả vận dụng lý thuyết vào đề tài

Sau gần 3 tuần tìm hiểu về lý thuyết và xây dựng cấu trúc giải thuật của chương trình thì có những quan trọng sau để trò chơi có thể chạy được

2.2.1. Khai báo thư viện

```
import pygame #Thư viện dùng để tạo các trò chơi với đồ họa 2D và âm thanh.
import sys #Thư viện để xử lý các thao tác hệ thống như thoát khỏi chương trình.
import random #Thư viện để màu sắc ngẫu nhiên cho trò chơi.
```

2.2.2. Khởi tạo Pygame và Pygame mixer

```
pygame.init() #Khởi tạo tất cả các mô-đun cần thiết của Pygame.
pygame.mixer.init() #Khởi tạo mô-đun âm thanh để phát âm thanh trong game.
```

2.2.3. Tải âm thanh

```
button_click_sound = pygame.mixer.Sound("button_click.wav") # Âm thanh khi nhấn nút
intro = pygame.mixer.Sound("intro.wav") # Âm thanh intro
```

```

win = pygame.mixer.Sound("win.wav") # Âm thanh khi người chơi thắng
lose = pygame.mixer.Sound("lose.wav") # Âm thanh khi người chơi thua
color_click_sound = pygame.mixer.Sound("color_click.wav") # Âm thanh khi
nhấn chọn màu

level_up_sound = pygame.mixer.Sound("level_up.wav") # Âm thanh khi lên
cấp

```

Các dòng trên dùng để tải các tệp âm thanh vào game. Những âm thanh này sẽ được phát tại các thời điểm nhất định trong trò chơi (ví dụ: khi người chơi thắng, thua, nhấn nút).

2.2.4. Khai báo các biến và hằng số

```

WIDTH, HEIGHT = 1000, 1000 # Kích thước cửa sổ trò chơi

COLORS = [
    (255, 0, 0), # Đỏ
    (0, 128, 0), # Xanh lá cây đậm
    (0, 0, 255), # Xanh dương
    (255, 165, 0), # Cam
    (255, 255, 0), # Vàng
    (128, 0, 128), # Tím
    (0, 255, 255) # Xanh da trời
]

```

WIDTH, HEIGHT là kích thước cửa sổ trò chơi.

Các hằng số này định nghĩa màu sắc sử dụng trong trò chơi.

2.2.5. Khởi tạo cửa sổ

```

screen = pygame.display.set_mode((WIDTH, HEIGHT)) # Tạo cửa sổ với kích
thước đã định nghĩa

pygame.display.set_caption("Trò Chơi Đoán Màu") # Đặt tiêu đề cửa sổ

```

2.2.6. Font chữ

Sử dụng font được thêm vào vì font chữ của chương trình khi viết tiếng Việt có dấu sẽ bị lỗi định dạng.

```

font_title = pygame.font.Font("InterTight-VariableFont_wght.ttf", 72) # Tiêu đề
font_button = pygame.font.Font("InterTight-VariableFont_wght.ttf", 36) # Nút bấm
font_text = pygame.font.Font("InterTight-Italic-VariableFont_wght.ttf", 28) # Chữ
thường

```

2.2.7. Các mục trong menu

`menu_items`: Một danh sách chứa các mục trong menu chính. Mỗi mục bao gồm tên hiển thị và mô tả về mục đó.

```
menu_items = [  
    ("GIỚI THIỆU", "Giới thiệu về trò chơi"),  
    ("HƯỚNG DẪN", "Hướng dẫn cách chơi"),  
    ("CHẠY DEMO", "Chạy demo trò chơi"),  
    ("THOÁT", "Thoát trò chơi")  
]
```

2.2.8. Hàm vẽ trò chơi

Hàm `ve_tro_choi` sẽ vẽ giao diện trò chơi, bao gồm các vòng màu, nút "Kết Quả", "Thoát", và các thông tin khác như số lần đoán, mức độ chơi, v.v..

```
def ve_tro_choi(guesses, results, current_guess, round_size, show_answer,  
secret_colors, level):
```

2.2.9. Hàm hiển thị game Over

Hàm này sẽ hiển thị màn hình "Game Over" khi người chơi thua, và hiển thị đáp án đúng của trò chơi.

```
def hien_thi_game_over(secret_colors, round_size):
```

2.2.10. Hàm vòng lặp trò chơi (Hàm quan trọng)

Đây là hàm chính điều khiển vòng lặp trò chơi, trong đó chương trình sẽ tạo màu ngẫu nhiên và sử dụng vòng lặp thông báo kết quả.

```
def vong_lap_trong_tro_choi():  
    global intro_played # Biến toàn cục intro  
    intro.stop() # Dừng âm thanh intro khi vào mục "Giới thiệu"  
    intro_played = False # Đặt lại intro_played  
    running = True # Theo dõi trạng thái của vòng lặp chính trò chơi  
    round_sizes = [4, 5, 6, 7] # Danh sách chứa số lượng màu đoán  
    round_index = 0 # Lưu chỉ số vòng chơi hiện tại  
    level = 1 # Gán giá trị cho level  
  
    secret_colors = random.sample(range(1, round_sizes[round_index] +  
1), round_sizes[round_index]) # Tạo danh sách màu ngẫu nhiên không trùng  
    guesses = [] # Lưu lượt đoán của người chơi  
    results = [] # Lưu kết quả số màu đoán đúng cho mỗi lượt
```

```

current_guess = [] # Lưu trữ các màu người chơi
attempts = 0 # Biến đếm số lượt đoán của người chơi
max_attempts = 10 # Số lượt đoán tối đa
show_answer = False # Biến để theo dõi trạng thái hiển thị đáp án

```

❖ **Đoạn mã kiểm tra người chơi nhập từ bàn phím:**

```

elif event.type == pygame.KEYDOWN:
    if event.key == pygame.K_ESCAPE: # Nhấn ESC để quay lại menu
        button_click_sound.play() # Phát âm thanh khi nhấn ESC
        main_menu() # Quay lại menu khi ESC được nhấn
        return
    if event.key in [pygame.K_1, pygame.K_2, pygame.K_3, pygame.K_4,
pygame.K_5, pygame.K_6, pygame.K_7]:
        color_index = event.key - pygame.K_1 + 1
        if color_index <= round_sizes[round_index]:
            current_guess.append(color_index)
            color_click_sound.play() # Phát âm thanh khi chọn

            if len(current_guess) == round_sizes[round_index]:
                guesses.append(current_guess)
                correct = sum(
                    [1 for i in range(round_sizes[round_index]) if
current_guess[i] == secret_colors[i]])
                results.append(correct)
                current_guess = []
                attempts += 1

```

Chức năng:

Đoạn mã này xử lý sự kiện nhấn phím từ người chơi. Nếu người chơi nhấn phím ESC, trò chơi sẽ quay lại menu chính. Nếu nhấn các phím số từ 1 đến 7, trò chơi sẽ thêm màu vào lượt đoán của người chơi.

Sau khi người chơi chọn đủ số màu cần đoán, kết quả sẽ được tính toán và lưu vào danh sách guesses và results.

Mỗi lượt đoán đều có phản hồi âm thanh khi người chơi chọn màu.

❖ **Đoạn mã kiểm tra lượt đoán của người chơi:**

```

while running:
    ve_tro_choi(guesses, results, current_guess, round_sizes[round_index],
show_answer, secret_colors, level)
    if attempts >= max_attempts:
        hien_thi_game_over(secret_colors)
        lose.play() # Phát âm thanh thua khi game over
        running = False

```

```

        continue

    if len(results) > 0 and results[-1] == round_sizes[round_index]:
        if round_index == len(round_sizes) - 1:
            result_text = font_title.render("You Win!", True, (0, 255,
0))
            screen.blit(result_text, (WIDTH // 2 -
result_text.get_width() // 2, HEIGHT // 2))
            win.play() # Phát âm thanh khi người chơi hoàn thành
toàn bộ trò chơi

            pygame.display.flip()
            pygame.time.delay(3000)
            running = False
        else:
            round_index += 1
            level += 1 # Tăng level khi chuyển sang vòng mới
            secret_colors = [random.randint(1,
secret_colors = random.sample(range(1,
round_sizes[round_index] + 1), round_sizes[round_index])

            guesses = []
            results = []
            current_guess = []
            attempts = 0
            show_answer = False

            level_up_sound.play

        continue

```

Giải thích:

Kiểm tra nếu người chơi đã đoán đúng hết màu trong lượt đoán hiện tại.

Kiểm tra xem người chơi đã hoàn thành tất cả các vòng hay chưa.

Nếu người chơi đã hoàn thành tất cả các vòng (vòng cuối), hiển thị thông báo "You Win!" và kết thúc trò chơi.

Nếu người chơi chưa hoàn thành tất cả các vòng, tăng cấp độ, tạo màu mới và khởi động lại lượt đoán cho vòng tiếp theo.

Tiếp tục vòng lặp trò chơi nếu chưa kết thúc.

2.2.11. Hàm hiển thị giới thiệu

Hàm này sẽ hiển thị nội dung giới thiệu về trò chơi khi người chơi chọn mục "Giới thiệu" trong menu.

```
def gioi_thieu():
```

2.2.12. Hàm hướng dẫn

Hàm này sẽ hiển thị các hướng dẫn cho người chơi về cách chơi trò chơi.

2.2.13. Hàm vẽ menu chính

Hàm này vẽ menu chính với các nút lựa chọn cho người chơi.

```
def ve_menu(selected_index):
```

2.2.14. Hàm quản lý menu

Hàm này quản lý menu chính và xử lý sự kiện khi người chơi nhấn vào các mục trong menu.

```
def main_menu():
```

2.2.15. Khởi chạy chương trình

Đây là phần bắt đầu chương trình. Khi chương trình được chạy, hàm `main_menu()` sẽ được gọi để hiển thị menu chính.

```
if __name__ == "__main__":  
    main_menu()
```

2.2.16. Thực thi demo

Đây là chương trình đầy đủ: vì chương trình quá dài nên không thể đưa lên hết được nên em đã thêm vào:

- [Đây là đường link Github](#)
- [Đây là đường link GDrive](#)
- Hoặc (<https://github.com/Khagn1702/TroChoiDoanMau>) và (<https://drive.google.com/drive/u/2/folders/177miXF4K3r7aZgr2HmEH6M0Ydm1qkRol>)

Chương 3: ỨNG DỤNG

3.1. Phân tích yêu cầu bài toán và xây dựng các cấu trúc dữ liệu

3.1.1. Phân tích yêu cầu

Ngữ cảnh bài toán:

Trò chơi "Đoán màu" mang tính thử thách logic, trong đó người chơi phải đoán đúng chuỗi màu bí mật được tạo ngẫu nhiên bởi máy tính.

Đây là bài toán liên quan đến xử lý dữ liệu đầu vào từ người chơi, so sánh với dữ liệu được máy tính tạo ra và cung cấp phản hồi.

Yêu cầu chức năng:

Máy tính tạo ngẫu nhiên chuỗi màu bí mật với số màu trong danh sách COLORS. Số lượng màu trong chuỗi bí mật (4) cố định, nhưng số lượng màu sẵn có (4-7) tăng theo độ khó.

Người chơi nhập chuỗi đoán (gồm 4 màu) thông qua các lựa chọn từ giao diện.

Sau mỗi lần đoán:

- Máy tính so sánh chuỗi đoán với chuỗi bí mật.
- Trả về số màu đoán đúng và đúng vị trí.
- Người chơi có tối đa 10 lượt đoán để tìm ra chuỗi màu bí mật.
- Trò chơi kết thúc khi người chơi đoán đúng hoặc hết lượt đoán:
- Nếu đoán đúng, hiển thị thông báo thắng.
- Nếu hết lượt đoán, hiển thị chuỗi màu bí mật và thông báo thua.

3.1.2. Cấu trúc dữ liệu

Danh sách màu sắc (COLORS):

Một danh sách chứa các màu sẵn có được định nghĩa theo định dạng RGB.

Ví dụ: [(255, 0, 0), (0, 255, 0), (0, 0, 255), ...].

Chuỗi màu bí mật (secret_colors):

Một danh sách chứa 4 màu được máy tính chọn ngẫu nhiên từ COLORS.

Ví dụ: [2, 4, 1, 3] (với 2 đại diện cho COLORS[1] là màu xanh lá cây đậm).

Lượt đoán (guesses):

Danh sách các lượt đoán của người chơi.

Mỗi phần tử là một danh sách con chứa 4 màu mà người chơi đã chọn.

Ví dụ: [[1, 3, 2, 4], [4, 2, 1, 3]].

Phản hồi (reusults):

Danh sách các phản hồi tương ứng với mỗi lượt đoán.

Mỗi phần tử là một số nguyên biểu thị số màu đoán đúng và đúng vị trí.

Ví dụ: [2, 3] (nghĩa là lượt đoán đầu tiên đoán đúng 2 màu đúng vị trí, lượt thứ hai đoán đúng 3 màu).

Biến hỗ trợ:

attempts: Số lần đoán hiện tại (giá trị tối đa là 10).

max_attempts: Giới hạn số lần đoán.

show_answer: Biến boolean xác định xem có hiển thị chuỗi bí mật không.

Mức độ khó (round_size):

Danh sách số lượng màu sẵn có tùy theo vòng chơi.

Ví dụ: [4, 5, 6, 7].

Các biến khác:

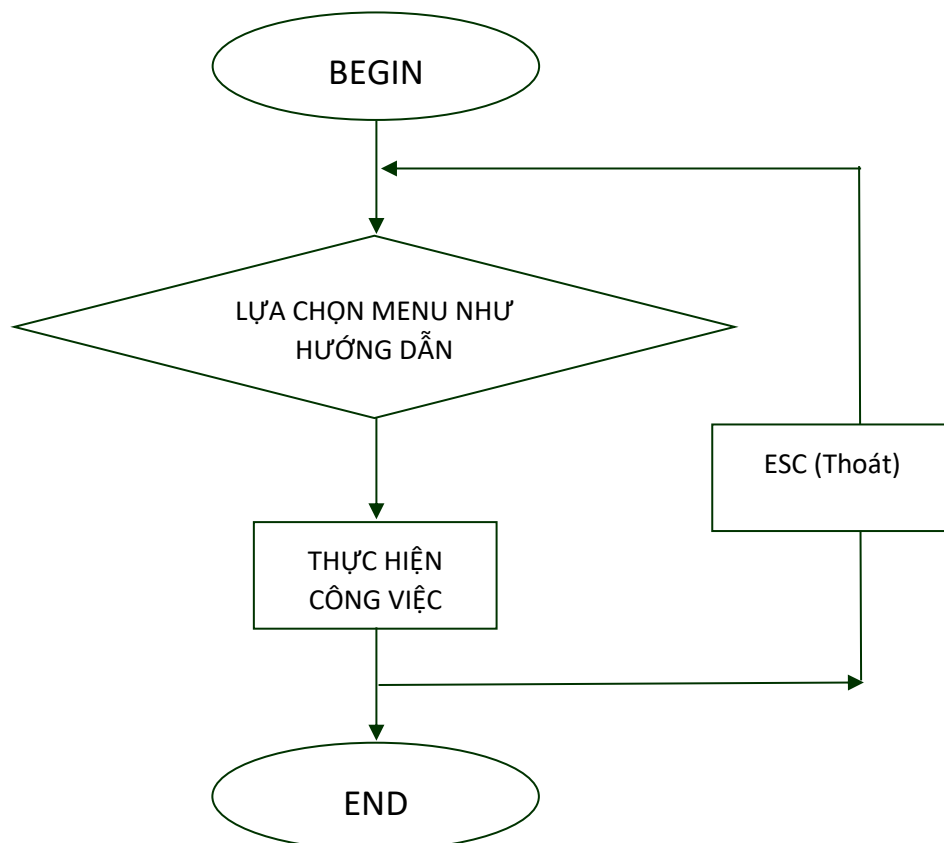
level: Theo dõi cấp độ hiện tại.

current_guess: Lưu chuỗi màu mà người chơi đang nhập.

3.2. Thiết kế giải thuật (Lưu đồ - Ngôn ngữ giả)

3.2.1. Lưu đồ và ngôn ngữ giả chương trình chính

Lưu đồ:



BẮT ĐẦU

Hiển thị menu chính với các lựa chọn:

1. Giới thiệu
2. Hướng dẫn
3. Chạy Demo
4. Thoát

Lựa chọn = Người chơi chọn một trong các mục trong menu.

Nếu lựa chọn == "Giới thiệu":

Hiển thị thông tin giới thiệu về trò chơi.

Nếu lựa chọn == "Hướng dẫn":

Hiển thị hướng dẫn cách chơi trò chơi.

Nếu lựa chọn == "Chạy Demo":

Bắt đầu trò chơi (game loop).

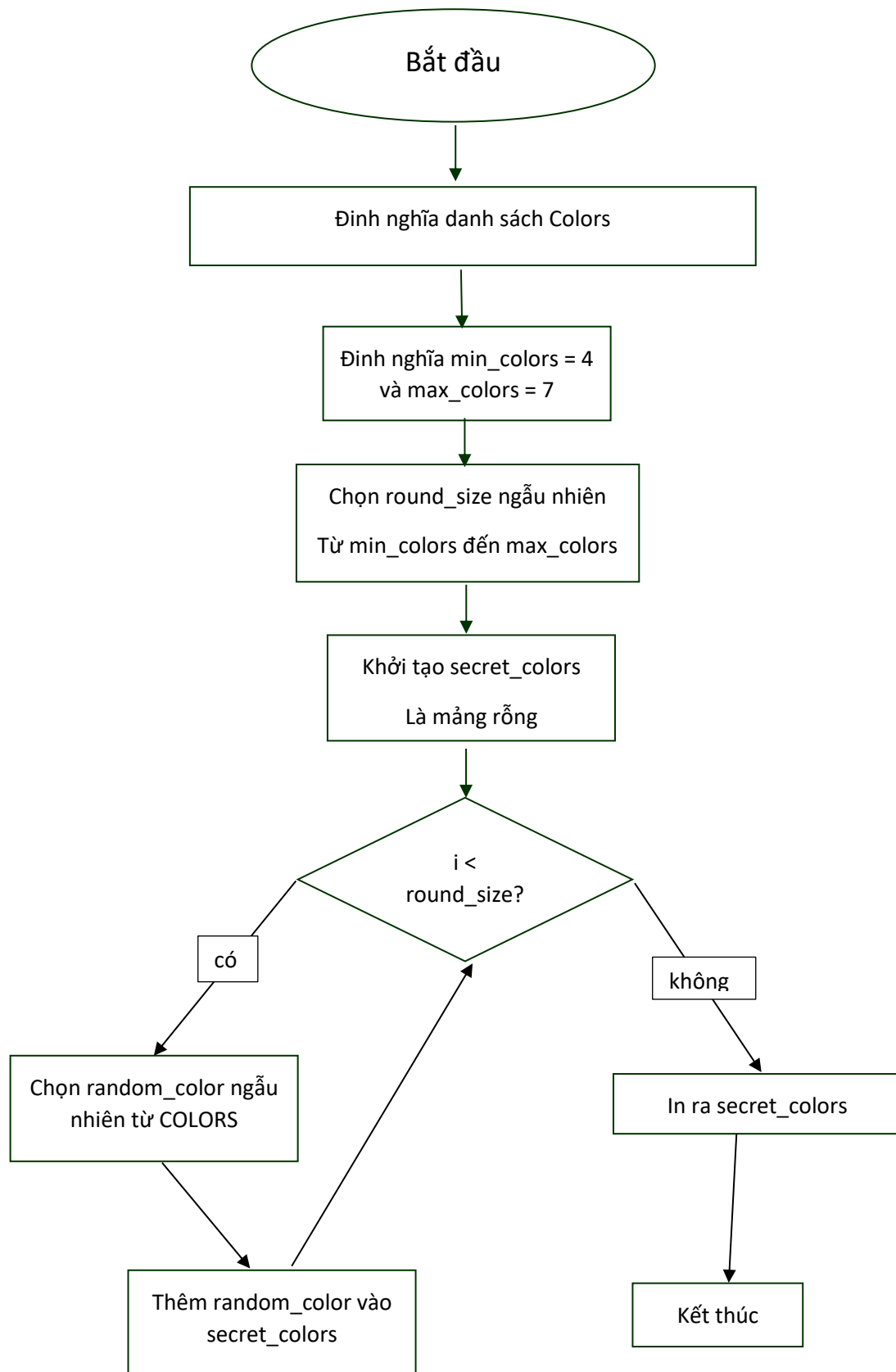
Nếu lựa chọn == "Thoát":

Thoát khỏi trò chơi.

KẾT THÚC

3.2.2. Lưu đồ và sơ đồ giả chương trình tạo màu ngẫu nhiên

Lưu đồ:



Sơ đồ giả:

BẮT ĐẦU

```
// Định nghĩa danh sách các màu có sẵn
COLORS = [Đỏ, Xanh lá, Xanh dương, Cam, Vàng, Tím, Xanh da trời]

// Định nghĩa số lượng màu tối thiểu và tối đa mà người chơi cần đoán
min_colors = 4
max_colors = 7

// Chọn số màu ngẫu nhiên trong phạm vi từ min_colors đến max_colors
round_size = Chọn số ngẫu nhiên trong khoảng từ min_colors đến max_colors

// Tạo một danh sách màu ngẫu nhiên cho máy
secret_colors = []

Lặp qua số lượng màu cần tạo (round_size):
    // Chọn một màu ngẫu nhiên từ danh sách COLORS
    random_color = Chọn ngẫu nhiên từ COLORS

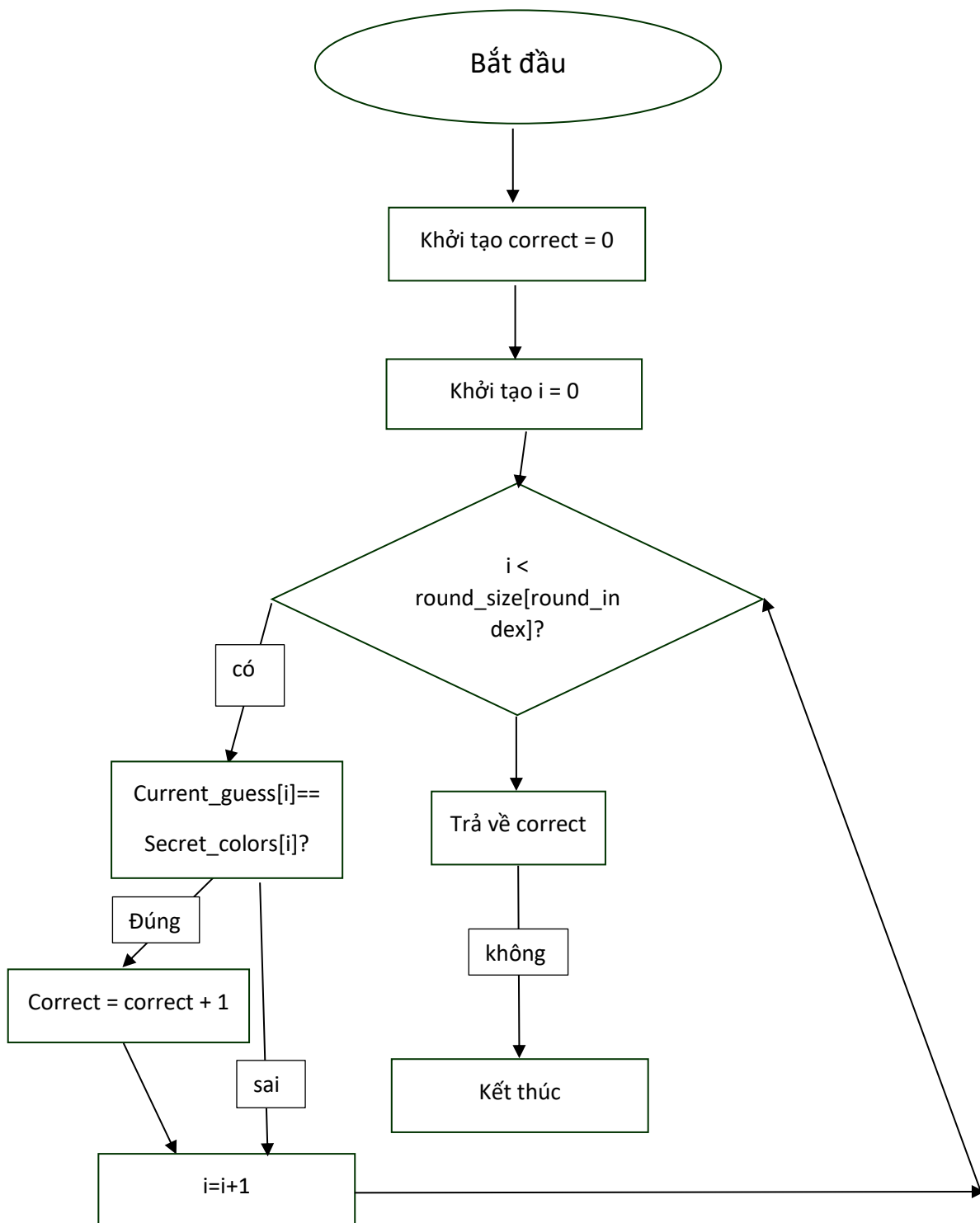
    // Thêm màu vào danh sách secret_colors
    Thêm random_color vào secret_colors

// Kết quả cuối cùng là danh sách các màu mà máy đã chọn
In ra secret_colors
```

KẾT THÚC

3.2.3. Lưu đồ và sơ đồ giả chương trình đoán màu:

Lưu đồ:



Sơ đồ giả:

Bắt đầu:

```
correct = 0 // Khởi tạo số lượng màu đúng
```

```
i = 0 // Khởi tạo chỉ số i
```

```
Trong khi i < round_sizes[round_index]: // Kiểm tra nếu còn màu để so sánh
```

```
    Nếu current_guess[i] == secret_colors[i]: // So sánh màu đoán với màu bí mật
```

```
        correct = correct + 1 // Nếu đúng, tăng số lượng màu đúng lên 1
```

```
    Kết thúc nếu
```

```
    i = i + 1 // Tăng chỉ số i để kiểm tra màu tiếp theo
```

```
    Kết thúc trong khi
```

```
    Trả về correct // Trả về tổng số màu đúng
```

```
    Kết thúc
```

3.3. Giới thiệu về chương trình

3.3.1 Menu chương trình

Khi chạy thì chương trình có giao diện như sau:



Để biết về trò chơi bạn chọn **GIỚI THIỆU** trên thanh Menu, giao diện như sau:

GIỚI THIỆU

ĐẶC TẢ ĐỀ TÀI

Nội dung bài toán:

“Đoản màu” được thể hiện: có từ 4 đến 7 màu, người chơi một lần đoán 4 màu. Máy sẽ đánh giá mỗi lần đoán và cho biết số vị trí đoán đúng. Sau 10 lần đoán máy sẽ cho kết quả người chơi thắng hay thua.

YÊU CẦU CỦA ĐỀ TÀI

Nắm vững cơ sở lý thuyết về cấu trúc dữ liệu. Các giải thuật sắp xếp. Chương trình cần có các chức năng sau đối với các giải thuật:

Đọc dữ liệu từ file văn bản, xuất ra file văn bản.

Cho biết số lần đổi chỗ của các giải thuật.

So sánh các giải thuật với nhau.

Nhấn ESC để quay lại menu

Để biết được cách chơi như thế nào bạn chọn **HƯỚNG DẪN** trên thanh Menu của giao diện như sau:

HƯỚNG DẪN

1. MÔ TẢ TRÒ CHƠI:

Trò chơi Đoán Màu yêu cầu người chơi đoán một chuỗi màu sắc từ 4 đến 7 màu. Mỗi lần người chơi sẽ đoán 4 màu, và trò chơi cho phép 10 lần đoán. Mỗi lần đoán, trò chơi sẽ phản hồi bằng cách cho biết số lượng màu sắc và vị trí đúng.

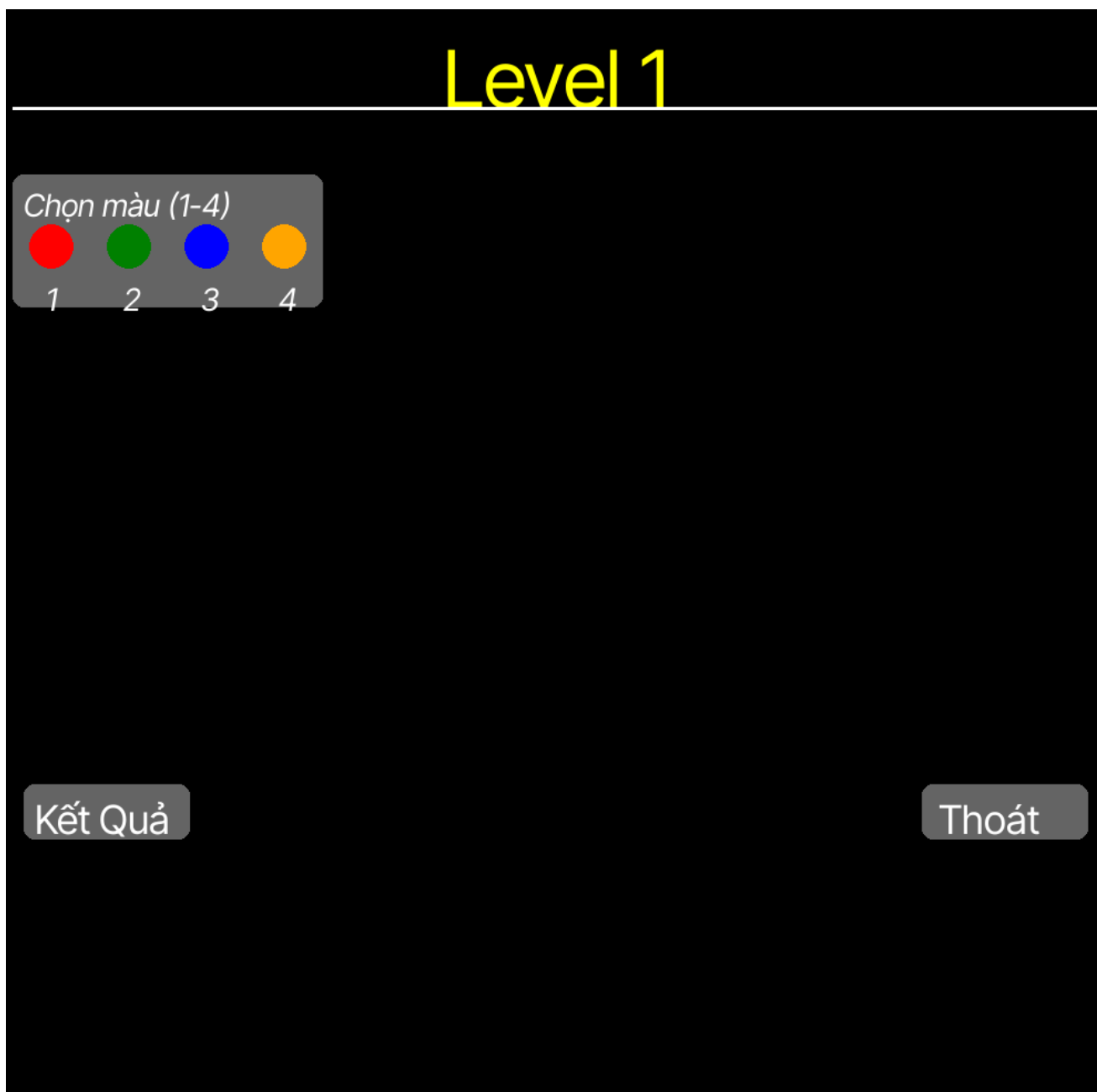
2. LUẬT CHƠI:

- Người chơi chỉ được chọn từ một danh sách các màu có sẵn.
- Mỗi lần đoán, bạn sẽ phải chọn 4 màu trong số các màu được cung cấp.
- Sau mỗi lần đoán, trò chơi sẽ phản hồi với số lượng màu và vị trí đúng.
- Người chơi có tổng cộng 10 lượt đoán để đoán chính xác chuỗi màu của máy.

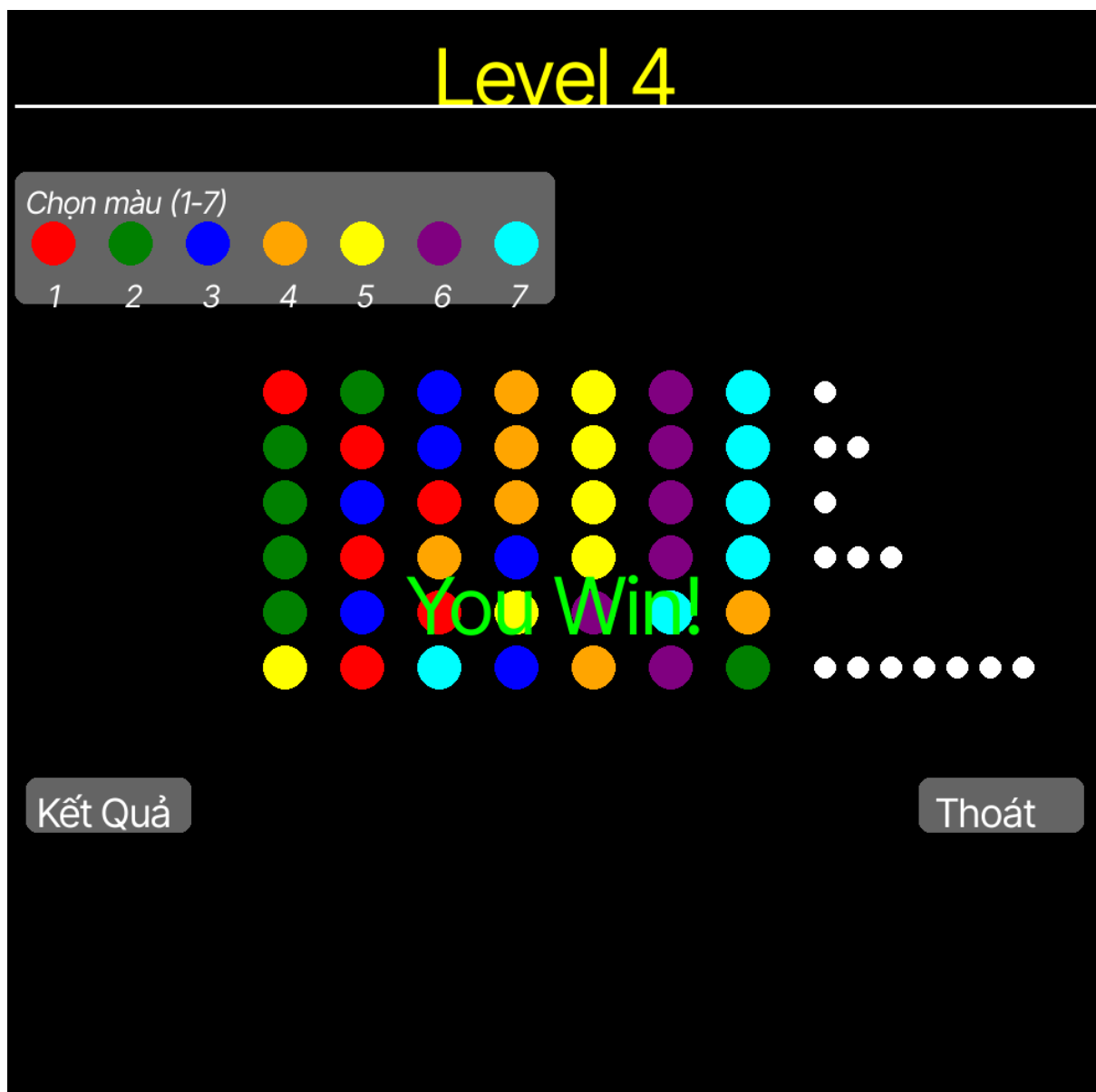
3. CÁCH CHƠI:

- Người chơi nhập màu từ bàn phím số (1 đến 7)
- Nhấn kết quả nếu như bạn không đoán được.

Để bắt đầu trò chơi bạn chọn **CHẠY DEMO** trên thanh Menu của, có giao diện như sau:



Khi bạn đoán đúng chương trình sẽ chạy tiếp các level khác như sau:



Trong hình nếu bạn đoán đúng thì chương trình thông báo kế bên bằng các chấm tròn nhỏ mỗi chấm tròn tương ứng với số màu bạn đoán đúng.

Khi bạn đoán đúng thì chương trình sẽ báo **YOU WIN** ở giữa màn hình:

Còn nếu sau 10 lượt bạn vẫn đoán không đúng thì chương trình Thông báo thua và cho bạn xem đáp án sau đó trở về menu chính:

Game Over!

Đáp án đúng:



CHƯƠNG 4: KẾT LUẬN

4.1. Nhận xét kết quả đạt được

- Cơ bản hoàn thành các yêu cầu đặt ra.
- Giao diện và âm thanh: Chương trình có giao diện đơn giản nhưng dễ hiểu và âm thanh hiệu quả (âm thanh có các hành động thắng, thua, nhạc intro, v.v.). Điều này giúp tăng tính tương tác và thú vị của trò chơi.
- Tăng vốn kiến thức về lập trình của em. Trong quá trình Em đã tham khảo rất nhiều ngôn ngữ cũng như là môi trường cho trường trình.
- Đây là loại trò chơi có thể dùng để giải trí trong những lúc căng thẳng.

4.2. Những hạn chế

Đây là niên luận đầu tiên của em nên cũng gặp rất nhiều khó khăn, tuy cơ bản đã hoàn thành được yêu cầu đề ra nhưng không tránh khỏi những hạn chế. Sau đây là số hạn chế do Em nhận thấy:

- **Hạn chế về độ khó:** Trò chơi hiện chỉ có một cấu trúc vòng chơi khá đơn giản, với tối đa 10 lượt đoán và không có các cấp độ khó khác ngoài việc thay đổi số lượng màu sắc cho phép đoán (từ 4 đến 7 màu). Điều này có thể khiến trò chơi thiếu thử thách đối với người chơi có kinh nghiệm.
- **Tính năng giới hạn:** Hiện tại, trò chơi chỉ có các tính năng cơ bản như giới thiệu, hướng dẫn và chơi demo mà không có thêm các tính năng thú vị khác như chế độ nhiều người chơi, tính điểm hay bảng xếp hạng.
- **Tương tác giới hạn:** Người chơi chỉ có thể chọn các màu bằng phím số (1-7) hoặc nhấp chuột. Mặc dù đơn giản nhưng có thể hạn chế sự đa dạng trong cách thức tương tác

Do thời gian có hạn nên trò chơi chưa được phát triển một cách hoàn hảo.

4.3. Hướng phát triển

Trong tương lai sẽ hoàn thành chương trình trò chơi một cách tốt nhất không còn một sai sót nào, nâng cấp thêm một số chức năng trong chương trình (Cải thiện độ khó, thêm tính năng thú vị vào game, hình ảnh đẹp hơn, tương tác với người chơi nhiều hơn...)

Những cải tiến này có thể làm cho trò chơi trở nên thú vị hơn và thu hút nhiều người chơi hơn trong tương lai.

PHỤ LỤC

Hướng dẫn cài đặt:

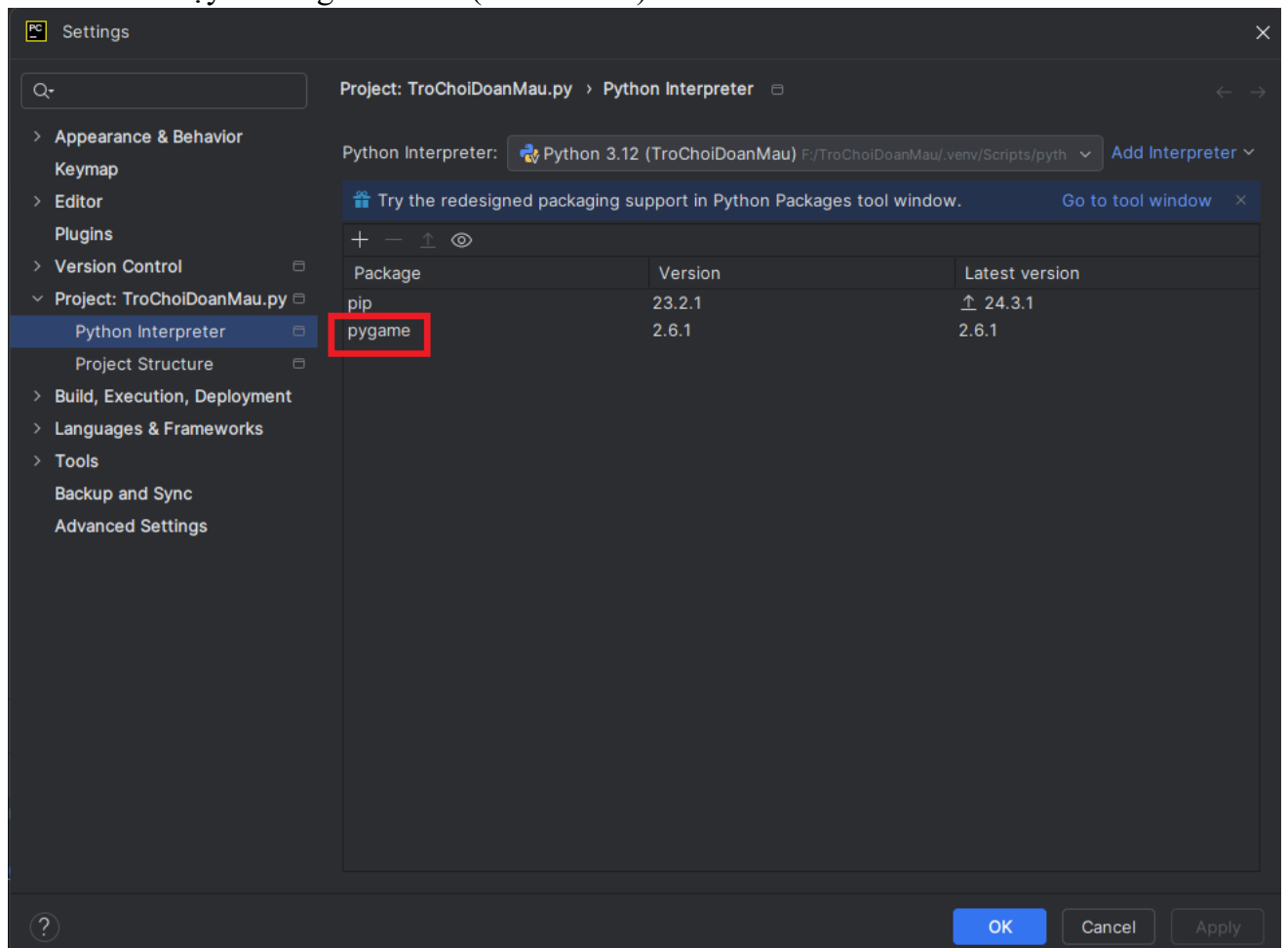
Bước 1: [Tải Python\(Pycharm\)](#)

Bước 2: Mở file *TroChoiDoanMau.py* lên

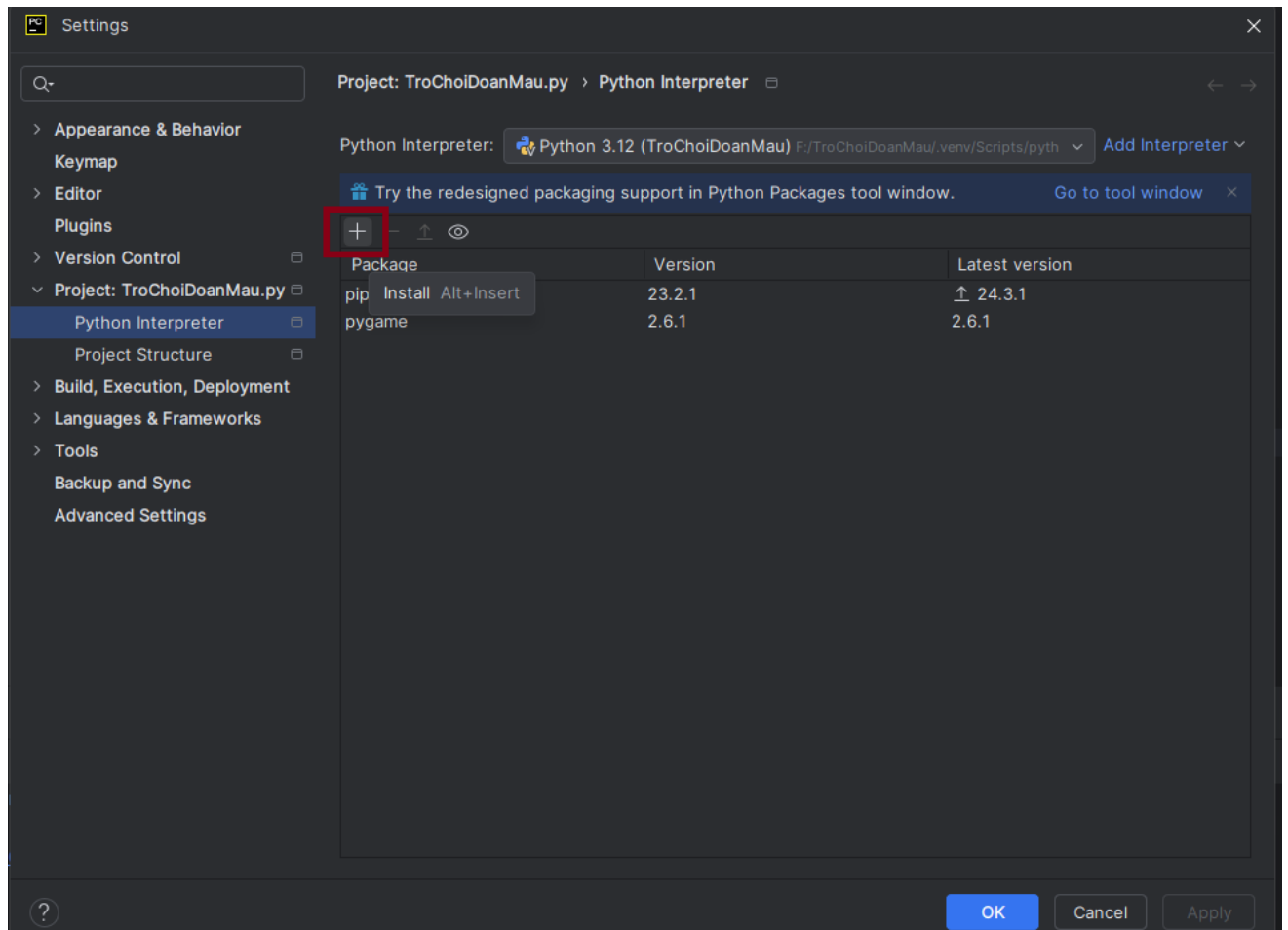
Bước 3: xem chương trình đã cài đặt thư viện Pygame chưa bằng cách sau:

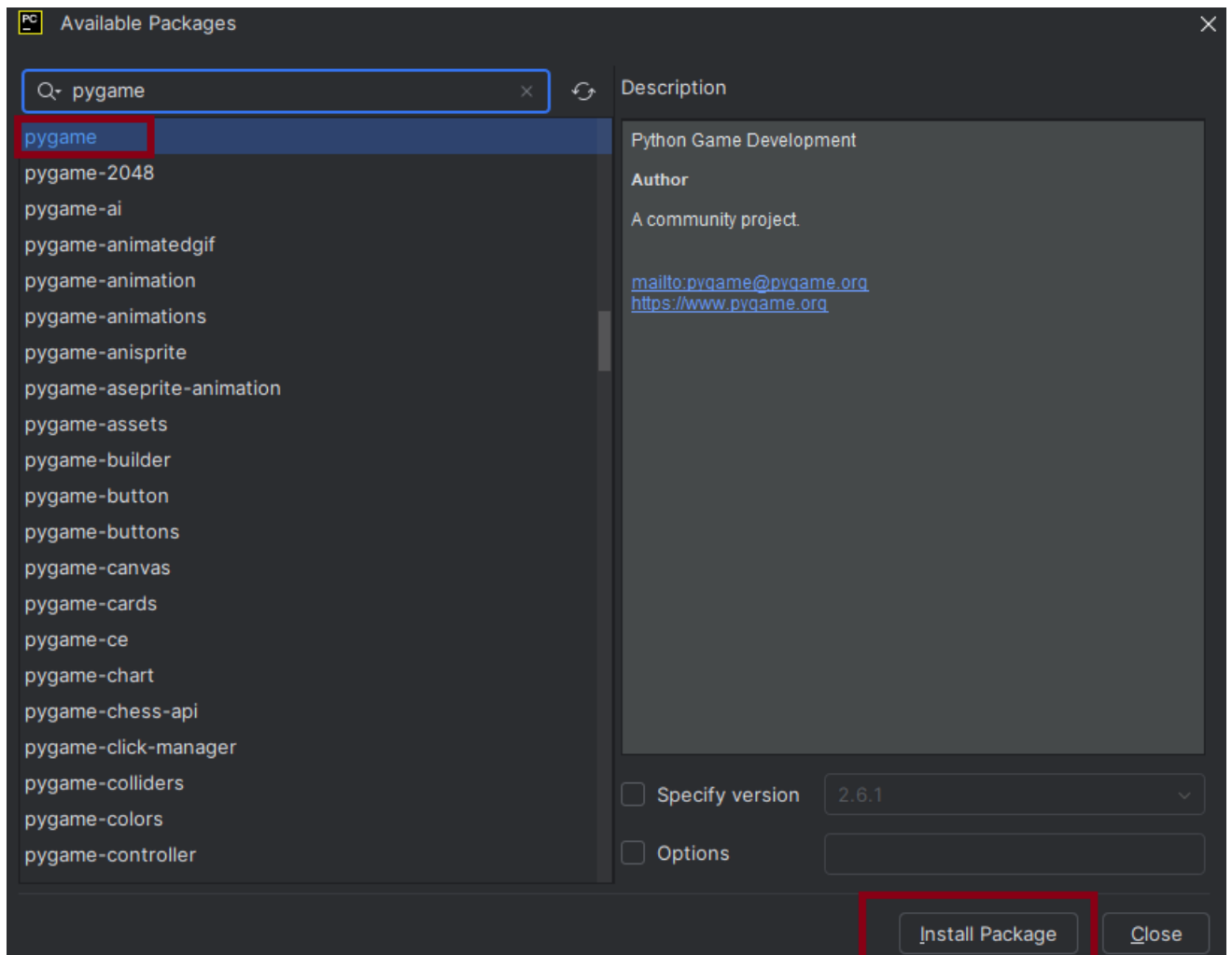
File->Settings->Project: TroChoiDoanMau.py

Click vào Python Interpreter sau đó xem đã có thư viện pygame chưa nếu có thì thoát ra và chạy chương trình Run(Shift + F10):



Nếu chưa cài thư viện Pygame thì bấm vào dấu cộng và tìm pygame về và Install Package để cài đặt thư viện





TÀI LIỆU THAM KHẢO

[1] Trần Phước Nghĩa

Bài giảng Lập trình Pascal, ĐH. Bạc Liêu.

[2] Ths. Trần Khanh Luân

Lập trình Python. Chương 2, Chương 4, Chương 5. ĐH Bạc Liêu, 2022

[3] Huỳnh Văn Mãi

Lập trình trò chơi đoán màu C++, ĐH Bạc Liêu

[4] Hà Văn Cường

Niên luận trò chơi đoán màu Pascal, ĐH Bạc Liêu