

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

Bộ môn: Công Nghệ Thông Tin.

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC

LẬP TRÌNH PYTHON

Sinh viên: Lê Quốc Trung

MSSV: k225480106065

Lớp: K58KTP

Giáo Viên Giảng Dạy: TS. Nguyễn Văn Huy

Link Github: <https://github.com/LeQuocTrung-810/BTL-Python>



Thái Nguyên – 2025

TRƯỜNG ĐHKTCN
KHOA ĐIỆN TỬ

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC: LẬP TRÌNH PYTHON

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Sinh viên: Lê Quốc Trung

MSSV: k225480106065

Lớp: K58KTP

Ngành: Kỹ thuật phần mềm

Giáo viên hướng dẫn: TS. Nguyễn Văn Huy

Ngày giao đề: 20/5/2025

Ngày hoàn thành: 10/6/2025

Tên đề tài: **Blackjack GUI đơn giản**

Yêu cầu: Xây game Blackjack (Chapter 9) với GUI: chia bài, hit/stand, tính điểm.

.....
.....
.....
.....

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày.....tháng.....năm 20...

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

MỤC LỤC

MỤC LỤC	1
LỜI CAM ĐOAN	2
LỜI NÓI ĐẦU.....	3
CHƯƠNG 1: GIỚI THIỆU ĐẦU BÀI	4
1.1. Mô tả đề bài.....	4
1.2. Tính năng yêu cầu.....	4
1.3. Thách thức	5
1.4. Kiến thức vận dụng.....	5
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	7
2.1. Luật chơi Blackjack cơ bản.....	7
2.2. Cấu trúc lập trình hướng đối tượng (OOP)	7
CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH.....	10
3.1. Cấu trúc tổng thể của hệ thống.....	10
3.2. Thiết kế lớp và chức năng	10
3.3. Luồng xử lý thuật toán chính	14
3.4. Giao diện đồ họa (GUI).....	16
3.5. Ưu điểm thiết kế.....	16
CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT LUẬN	18
4.1. Đánh giá kết quả đạt được	18
4.2. Hạn chế	18
4.3. Hướng phát triển.....	18
4.4. Kết luận.....	19
KẾT LUẬN	20
TÀI LIỆU THAM KHẢO	21

LỜI CAM ĐOAN

Em xin cam đoan rằng bài báo cáo với đề tài “*Blackjack GUI đơn giản*” là kết quả của quá trình học tập và nghiên cứu nghiêm túc của cá nhân em. Toàn bộ nội dung trong báo cáo đều do em tự thực hiện, không sao chép của người khác. Các tài liệu tham khảo (nếu có) đều được trích dẫn rõ ràng và đúng quy định. Em hoàn toàn chịu trách nhiệm trước kết quả và tính trung thực của nội dung bài báo cáo này.

Thái Nguyên, ngày...tháng...năm 2025

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ hiện nay, lập trình không chỉ là một kỹ năng quan trọng trong ngành công nghệ thông tin mà còn là công cụ sáng tạo giúp giải quyết các vấn đề thực tiễn và xây dựng các ứng dụng trực quan, thân thiện với người dùng. Việc học lập trình Python – một ngôn ngữ đơn giản, dễ đọc và linh hoạt – giúp em tiếp cận nhanh với tư duy lập trình hiện đại và phát triển phần mềm hiệu quả.

Đề tài “*Blackjack GUI đơn giản*” là một bài tập thực hành giúp em củng cố và vận dụng các kiến thức đã học về lập trình hướng đối tượng, xử lý giao diện đồ họa và quản lý sự kiện trong Python. Thông qua việc phát triển một trò chơi nhỏ nhưng đầy đủ chức năng như Blackjack, em có cơ hội hiểu rõ hơn cách tổ chức chương trình, tương tác người dùng và mô phỏng các quy tắc logic trong một ứng dụng thực tế.

Trong quá trình thực hiện đề tài, em đã gặp không ít khó khăn như thiết kế giao diện với tkinter, xử lý điểm của quân Át theo luật Blackjack, hay đảm bảo sự mượt mà khi giao diện cập nhật theo từng hành động của người chơi. Tuy nhiên, nhờ vào kiến thức tích lũy từ môn học, sự hướng dẫn tận tình của giảng viên và các tài liệu tham khảo, em đã dần hoàn thiện sản phẩm một cách có hệ thống và đạt được mục tiêu đề ra.

Báo cáo này trình bày chi tiết các bước thực hiện đề tài, từ phân tích yêu cầu, thiết kế hệ thống, đến cài đặt và đánh giá kết quả. Em xin chân thành cảm ơn sự hỗ trợ của thầy cô và mong nhận được những góp ý quý báu để có thể tiếp tục hoàn thiện kỹ năng lập trình và phát triển ứng dụng trong tương lai.

CHƯƠNG 1: GIỚI THIỆU ĐẦU BÀI

1.1. Mô tả đề bài

Trong khuôn khổ môn học lập trình Python, đề tài này yêu cầu xây dựng một trò chơi Blackjack đơn giản với giao diện đồ họa (Graphical User Interface – GUI). Đây là một trò chơi bài phổ biến, còn được biết đến với tên gọi “21 điểm”. Mục tiêu chính của người chơi là rút các lá bài sao cho tổng điểm gần 21 nhất nhưng không vượt quá con số này.

Trò chơi sẽ được xây dựng dựa trên kiến thức từ Chương 9 trong giáo trình, trong đó hướng dẫn cách tổ chức trò chơi theo hướng đối tượng và cách tích hợp giao diện người dùng. Các chức năng cơ bản cần được hiện thực bao gồm:

- Chia bài (Deal): Khi bắt đầu ván chơi, mỗi bên (người chơi và nhà cái) được chia hai lá bài.
- Rút bài (Hit): Người chơi có thể rút thêm bài để tăng điểm số nếu tổng điểm hiện tại chưa vượt quá 21.
- Dừng rút bài (Stand): Khi người chơi cảm thấy đủ điểm, họ có thể chọn dừng để chờ kết quả so điểm với nhà cái.
- Tính điểm và xác định thắng/thua: Sau khi người chơi dừng, nhà cái thực hiện lượt của mình và so sánh điểm với người chơi để xác định người thắng.

1.2. Tính năng yêu cầu

Trò chơi cần đáp ứng một số yêu cầu cơ bản để đảm bảo tính hoàn thiện và thân thiện với người dùng, bao gồm:

- Giao diện đồ họa trực quan: Sử dụng thư viện tkinter để tạo ra giao diện với các nút chức năng (Deal, Hit, Stand) và khu vực hiển thị các lá bài cho người chơi và nhà cái.
- Xử lý logic trò chơi đúng luật Blackjack: Bao gồm cách tính điểm, luật đặc biệt của quân Át (Ace có thể là 1 hoặc 11 điểm), và quy tắc dừng rút bài của nhà cái (dealer).
- Cập nhật trạng thái theo thời gian thực: Giao diện cần phản hồi ngay lập tức khi người chơi nhấn các nút hành động, đồng thời cập nhật điểm số, thông báo kết quả, và kiểm soát luồng trò chơi.

1.3. Thách thức

Trong quá trình xây dựng trò chơi, sinh viên có thể gặp phải một số khó khăn, bao gồm:

- Thiết kế giao diện với tkinter: Việc sắp xếp bố cục, cập nhật hình ảnh và xử lý sự kiện tương tác yêu cầu sự hiểu biết rõ ràng về thư viện tkinter.
- Xử lý logic quân Át: Trong Blackjack, quân Át có thể được tính là 1 hoặc 11 tùy theo tình huống. Việc viết thuật toán để xử lý điều này đòi hỏi sự cẩn thận và linh hoạt.
- Cập nhật giao diện theo hành động: Khi người chơi chọn "Hit" hoặc "Stand", giao diện cần phải hiển thị ngay lá bài mới và điểm số, đồng thời không làm gián đoạn trải nghiệm người dùng.

1.4. Kiến thức vận dụng

Để hoàn thành đề tài này, người thực hiện cần vận dụng tổng hợp nhiều kỹ năng và kiến thức đã học:

- Lập trình hướng đối tượng (OOP): Tổ chức mã nguồn theo các lớp như Card, Deck, Hand, Game giúp mã rõ ràng và dễ bảo trì.
- Xử lý sự kiện GUI (Event-driven programming): Sử dụng cơ chế callback trong tkinter để xử lý các nút bấm và cập nhật giao diện theo tương tác người dùng.
- Cấu trúc dữ liệu: Các danh sách (list) để lưu trữ bộ bài và lá bài trên tay, dictionary để quản lý điểm số hoặc thông tin bổ sung.
- Thư viện tkinter: Đây là thư viện GUI tiêu chuẩn trong Python, dùng để xây dựng cửa sổ ứng dụng, hiển thị các nút điều khiển và các phần tử giao diện khác.

Tóm tắt chương

Chương này giới thiệu tổng quan về đề tài “Xây dựng trò chơi Blackjack GUI đơn giản”. Đây là một bài tập thực hành trong môn học lập trình Python, với mục tiêu phát triển một trò chơi Blackjack cơ bản có giao diện đồ họa sử dụng thư viện tkinter. Trò chơi cần có các chức năng chính như chia bài, rút bài, dừng rút và tính điểm theo đúng luật chơi Blackjack. Bên cạnh đó, hệ thống phải hiển thị các lá bài

và cập nhật điểm số theo thời gian thực khi người chơi tương tác. Trong quá trình thực hiện, em đã gặp một số thách thức như thiết kế giao diện bằng tkinter, xử lý logic đặc biệt của quân Át (Ace) với hai giá trị 1 và 11, cũng như đảm bảo giao diện phản hồi linh hoạt với các hành động của người chơi. Để hoàn thành đề tài, em đã vận dụng nhiều kiến thức như lập trình hướng đối tượng, xử lý sự kiện trong GUI, sử dụng các cấu trúc dữ liệu cơ bản như list và dictionary, cùng với việc áp dụng thư viện tkinter để xây dựng và điều khiển giao diện người dùng.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Luật chơi Blackjack cơ bản

Blackjack (hay còn gọi là “21 điểm”) là một trò chơi bài phổ biến trong các sòng bạc và được đơn giản hóa để phù hợp với các trò chơi lập trình. Mỗi ván bài bắt đầu với việc người chơi và nhà cái (dealer) được chia hai lá bài. Mục tiêu của trò chơi là có tổng điểm các lá bài càng gần 21 càng tốt nhưng không được vượt quá con số này.

- **Cách tính điểm:**
 - Các lá bài từ 2 đến 10 được tính theo đúng số ghi trên mặt bài.
 - Các lá J (Jack), Q (Queen), và K (King) đều được tính là 10 điểm.
 - Quân Át (Ace) là lá bài đặc biệt có thể được tính là **1 hoặc 11**, tùy thuộc vào tình huống có lợi cho người chơi nhất (sao cho tổng điểm không vượt quá 21).
- **Các hành động chính của người chơi:**
 - **Hit:** Rút thêm một lá bài để tăng tổng điểm.
 - **Stand:** Dừng rút bài và chuyển lượt cho nhà cái.
- **Luật của nhà cái:** Sau khi người chơi chọn "Stand", nhà cái sẽ rút bài liên tục cho đến khi tổng điểm của nhà cái lớn hơn hoặc bằng 17.
- **Xác định kết quả:**
 - Nếu người chơi có tổng điểm vượt quá 21 → thua (bust).
 - Nếu nhà cái vượt quá 21 → người chơi thắng.
 - Nếu cả hai đều không vượt quá 21 → ai có điểm cao hơn là người thắng.
 - Nếu hai bên bằng điểm → hòa (push).

2.2. Cấu trúc lập trình hướng đối tượng (OOP)

Để tổ chức mã lệnh một cách rõ ràng và dễ quản lý, trò chơi được xây dựng theo hướng lập trình hướng đối tượng (Object-Oriented Programming – OOP). Các thành phần chính của trò chơi được biểu diễn thông qua các lớp như sau:

- **BJ_Card:** Đại diện cho một lá bài đơn lẻ. Mỗi lá bài có hai thuộc tính cơ bản: **hạng (rank)** và **chất (suit)**.
- **BJ_Deck:** Đại diện cho bộ bài gồm 52 lá. Lớp này có chức năng tạo bộ bài đầy đủ, trộn bài (shuffle) và chia bài (deal).
- **BJ_Hand:** Đại diện cho tập hợp các lá bài của một người chơi hoặc nhà cái. Lớp này cung cấp chức năng tính điểm, thêm bài và kiểm tra điều kiện thắng/thua.
- **BJ_Game:** Quản lý toàn bộ quá trình của trò chơi. Lớp này xử lý việc bắt đầu ván mới, hành động của người chơi, lượt của nhà cái, tính toán kết quả và cập nhật giao diện.

Việc tách các chức năng thành các lớp rõ ràng giúp chương trình dễ mở rộng, dễ bảo trì và phản ánh đúng cách tổ chức logic của một trò chơi thực tế.

2.3. Thư viện tkinter

Để xây dựng giao diện người dùng (GUI), trò chơi sử dụng thư viện tkinter, là thư viện GUI tiêu chuẩn đi kèm với Python. Các thành phần chính của giao diện bao gồm:

- **Cửa sổ chính (Tk):** Khởi tạo giao diện và tạo khung hiển thị toàn bộ trò chơi.
- **Nút bấm (Button):** Dùng để tạo các nút điều khiển như "Deal", "Hit", "Stand". Khi người dùng nhấn vào nút, chương trình sẽ gọi hàm xử lý tương ứng bằng cách sử dụng tham số command.
- **Nhãn (Label):** Hiển thị thông tin như điểm số, kết quả thắng/thua, thông báo trạng thái.
- **Canvas (Canvas):** Dùng để hiển thị hình ảnh hoặc biểu diễn các lá bài bằng cách vẽ trực tiếp hoặc sử dụng ảnh.

Ngoài ra, tkinter còn cho phép xử lý sự kiện theo thời gian thực. Ví dụ: khi người chơi nhấn nút “Hit”, giao diện sẽ lập tức cập nhật để hiển thị lá bài mới và điểm số hiện tại.

Việc sử dụng tkinter giúp trò chơi trở nên sinh động, dễ tương tác và thân thiện với người dùng.

Tóm tắt chương

Chương này trình bày các cơ sở lý thuyết cần thiết để xây dựng trò chơi Blackjack có giao diện đồ họa. Trò chơi tuân theo luật Blackjack cơ bản: mỗi người chơi bắt đầu với hai lá bài và cố gắng đạt tổng điểm gần 21 nhất mà không vượt quá. Người chơi có thể chọn rút thêm bài (Hit) hoặc dừng lại (Stand), trong khi nhà cái sẽ tự động rút bài cho đến khi đạt ít nhất 17 điểm. Quân Át là lá bài đặc biệt có thể tính là 1 hoặc 11 tùy tình huống. Về mặt lập trình, chương trình được thiết kế theo hướng đối tượng với các lớp chính như BJ_Card (đại diện cho một lá bài), BJ_Deck (bộ bài), BJ_Hand (bài của người chơi hoặc nhà cái), và BJ_Game (quản lý toàn bộ trò chơi). Giao diện đồ họa được xây dựng bằng thư viện tkinter, sử dụng các thành phần như cửa sổ chính (Tk), nút (Button), nhãn (Label) và khung vẽ (Canvas) để hiển thị bài và xử lý tương tác người dùng theo thời gian thực.

CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. Cấu trúc tổng thể của hệ thống

Chương trình được chia thành ba module chính, mỗi module đảm nhiệm một vai trò riêng biệt:

- **cards.py**: Xử lý cấu trúc của lá bài và bộ bài.
- **games.py**: Quản lý logic của trò chơi và xử lý điểm số cho người chơi và nhà cái.
- **main.py**: Giao diện người dùng (GUI) sử dụng thư viện tkinter, là nơi người chơi tương tác với trò chơi.

Cách chia module như vậy giúp tách biệt rõ ràng giữa logic trò chơi và giao diện hiển thị, từ đó dễ dàng mở rộng và bảo trì.

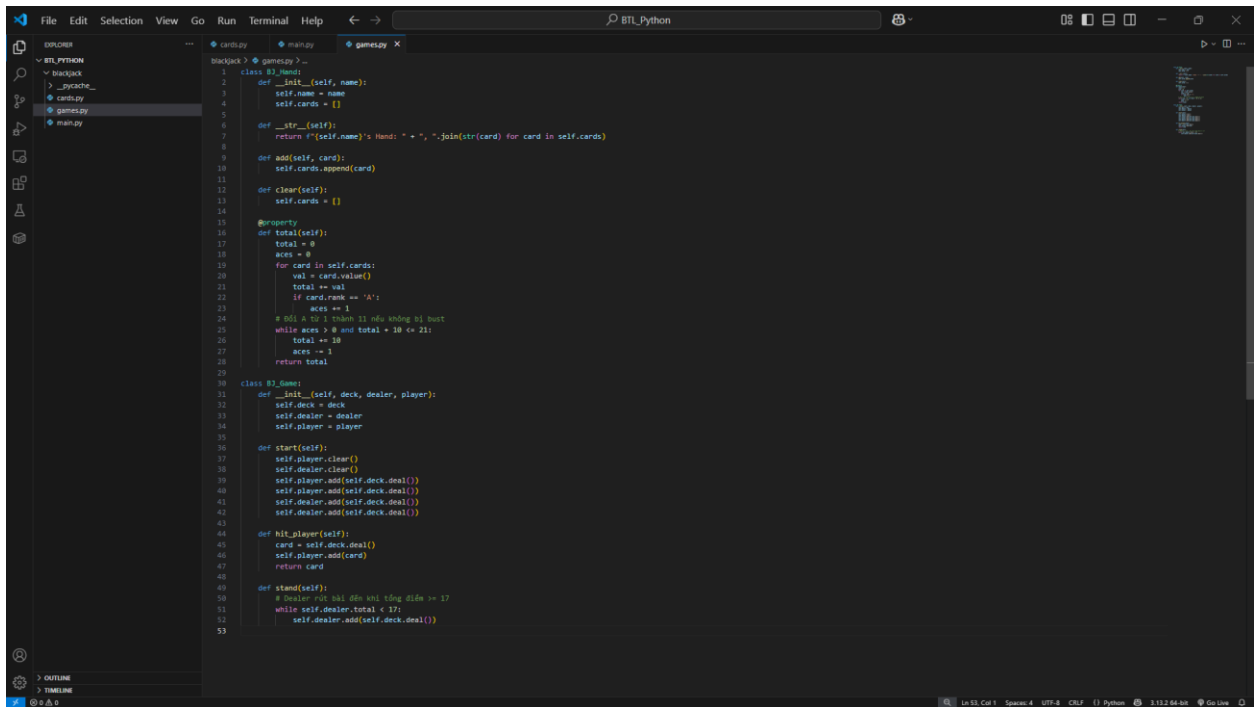
3.2. Thiết kế lớp và chức năng

a. cards.py

```
1 import random
2
3 class Card:
4     RANKS = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K']
5     SUITS = ['♠', '♣', '♥', '♦']
6
7     def __init__(self, rank, suit):
8         self.rank = rank
9         self.suit = suit
10
11     def __str__(self):
12         return f'{self.rank}{self.suit}'
13
14 class BJ_Card(Card):
15     def value(self):
16         if self.rank in ['J', 'Q', 'K']:
17             return 10
18         elif self.rank == 'A':
19             return 1
20         else:
21             return int(self.rank)
22
23 class BJ_Deck:
24     def __init__(self):
25         self.cards = [BJ_Card(rank, suit) for rank in Card.RANKS for suit in Card.SUITS]
26
27     def shuffle(self):
28         random.shuffle(self.cards)
29
30     def deal(self):
31         return self.cards.pop() if self.cards else None
32
```

- **Lớp Card:** Đại diện cho một lá bài với hai thuộc tính rank (A, 2–10, J, Q, K) và suit (♠, ♣, ♥, ♦).
- **Lớp BJ_Card:** Kế thừa từ Card, bổ sung phương thức value() để trả về điểm số tương ứng của lá bài theo luật Blackjack.
- **Lớp BJ_Deck:** Tạo bộ bài 52 lá (gồm các BJ_Card), có phương thức shuffle() để xáo bài và deal() để rút lá bài khỏi bộ.

b. games.py



```
1 class BJ_Hand:
2     def __init__(self, name):
3         self.name = name
4         self.cards = []
5
6     def __str__(self):
7         return f"{self.name}'s Hand: " + ", ".join(str(card) for card in self.cards)
8
9     def add(self, card):
10        self.cards.append(card)
11
12    def clear(self):
13        self.cards = []
14
15    @property
16    def total(self):
17        total = 0
18        aces = 0
19        for card in self.cards:
20            val = card.value
21            total += val
22            if card.rank == 'A':
23                aces += 1
24        # Nếu A từ 1 thành 11 nếu không bị bust
25        while aces > 0 and total + 10 <= 21:
26            total += 10
27            aces -= 1
28        return total
29
30 class BJ_Game:
31     def __init__(self, deck, dealer, player):
32         self.deck = deck
33         self.dealer = dealer
34         self.player = player
35
36     def start(self):
37         self.player.clear()
38         self.dealer.clear()
39         self.player.add(self.deck.deal())
40         self.player.add(self.deck.deal())
41         self.dealer.add(self.deck.deal())
42         self.dealer.add(self.deck.deal())
43
44     def hit_player(self):
45         card = self.deck.deal()
46         self.player.add(card)
47         return card
48
49     def stand(self):
50         # Dealer rút bài đến khi tổng điểm >= 17
51         while self.dealer.total < 17:
52             self.dealer.add(self.deck.deal())
53
```

- **Lớp BJ_Hand:** Đại diện cho "tay bài" của người chơi hoặc nhà cái.
 - Có phương thức `add()` để thêm bài, `clear()` để xóa bài.
 - Thuộc tính `total` là tổng điểm, có xử lý đặc biệt với quân Át (A có thể là 1 hoặc 11).
- **Lớp BJ_Game:** Kết nối bộ bài và hai tay bài. Xử lý các hành động chính của trò chơi như:
 - `start()`: chia bài ban đầu cho người chơi và nhà cái.
 - `hit_player()`: người chơi rút thêm bài.
 - `stand()`: nhà cái tự động rút bài đến khi đạt ≥ 17 điểm.

c. main.py

```

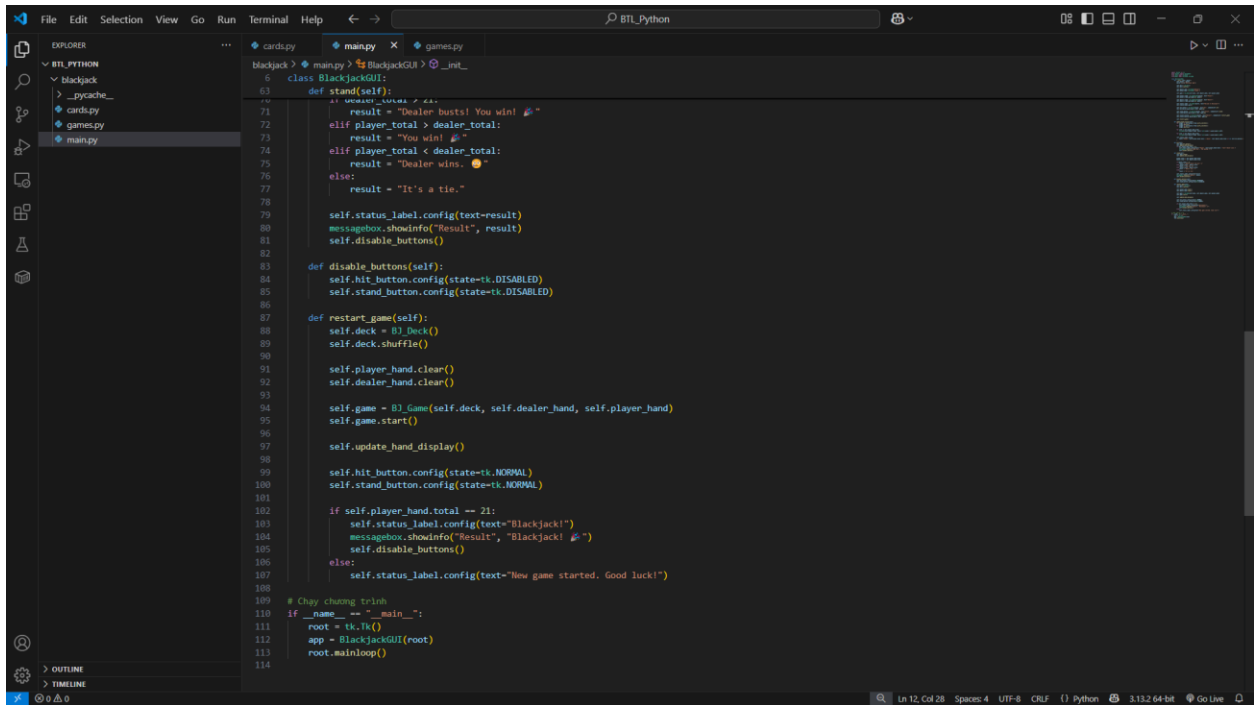
1 import tkinter as tk
2 from tkinter import messagebox
3 from cards import BJ_Deck
4 from games import BJ_Hand, BJ_Game
5
6 class BlackjackGUI:
7     def __init__(self, master):
8         self.master = master
9         master.title("Blackjack Game")
10
11         self.deck = BJ_Deck()
12         self.deck.shuffle()
13
14         self.dealer_hand = BJ_Hand("Dealer")
15         self.player_hand = BJ_Hand("Player")
16
17         self.game = BJ_Game(self.deck, self.dealer_hand, self.player_hand)
18
19         self.player_frame = tk.LabelFrame(master, text="Player")
20         self.player_frame.pack(padx=10, pady=5)
21
22         self.dealer_frame = tk.LabelFrame(master, text="Dealer")
23         self.dealer_frame.pack(padx=10, pady=5)
24
25         self.status_label = tk.Label(master, text="Welcome to Blackjack!")
26         self.status_label.pack()
27
28         self.hit_button = tk.Button(master, text="Hit", command=self.hit)
29         self.hit_button.pack(side=tk.LEFT, padx=10)
30
31         self.stand_button = tk.Button(master, text="Stand", command=self.stand)
32         self.stand_button.pack(side=tk.LEFT, padx=10)
33
34         self.restart_button = tk.Button(master, text="Restart", command=self.restart_game)
35         self.restart_button.pack(side=tk.LEFT, padx=10)
36
37         self.restart_game()
38
39     def update_hand_display(self):

```

```

39     def update_hand_display(self):
40         for widget in self.player_frame.winfo_children():
41             widget.destroy()
42         for widget in self.dealer_frame.winfo_children():
43             widget.destroy()
44
45         for card in self.player_hand.cards:
46             tk.Label(self.player_frame, text=f"{str(card)}").pack(side=tk.LEFT)
47
48         for card in self.dealer_hand.cards:
49             tk.Label(self.dealer_frame, text=f"{str(card)}").pack(side=tk.LEFT)
50
51         self.status_label.config(text=f"Player: {self.player_hand.total} | Dealer: {self.dealer_hand.total if not self.hit_button['state'] == tk.NORMAL else ' '}")
52
53     def hit(self):
54         self.game.hit_player()
55         self.update_hand_display()
56         if self.player_hand.total > 21:
57             self.status_label.config(text=f"Player: {self.player_hand.total} - Bust! Dealer wins.")
58             messagebox.showinfo("Game Over", "You busted! 🚫")
59             self.disable_buttons()
60
61     def stand(self):
62         self.game.stand()
63         self.update_hand_display()
64
65         player_total = self.player_hand.total
66         dealer_total = self.dealer_hand.total
67
68         if dealer_total > 21:
69             result = "Dealer busts! You win! 🎉"
70         elif player_total > dealer_total:
71             result = "You win! 🎉"
72         elif player_total < dealer_total:
73             result = "Dealer wins. 😞"
74         else:
75             result = "Push! 🤝"
76         messagebox.showinfo("Game Over", result)

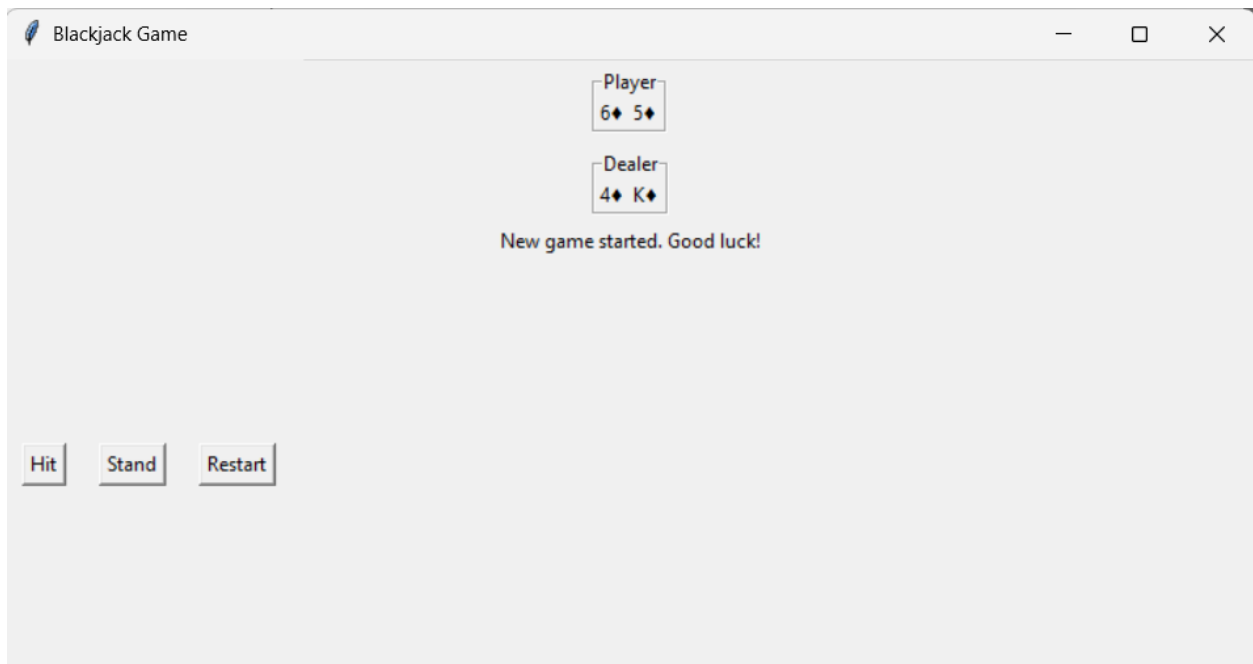
```

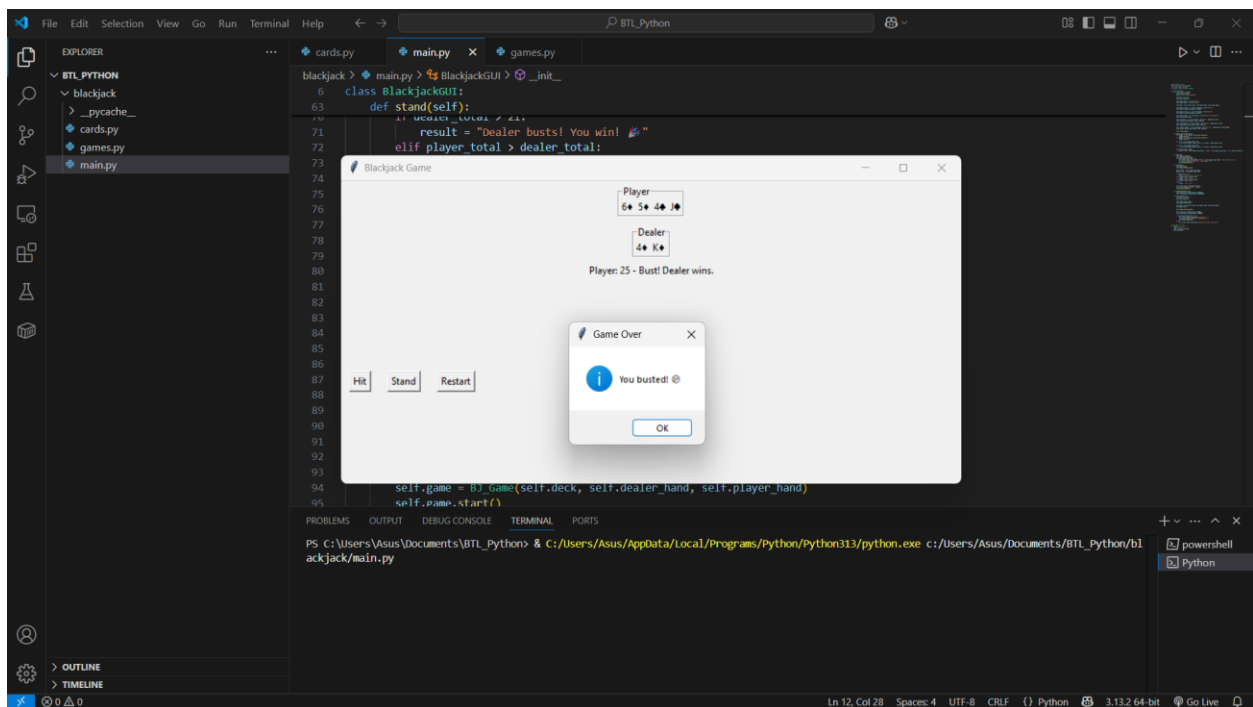
- **Lớp BlackjackGUI:** Là giao diện đồ họa chính.
 - Tạo khung hiển thị bài của người chơi và nhà cái.
 - Hiển thị điểm số và thông báo kết quả.
 - Cung cấp 3 nút: **Deal/Restart, Hit, Stand.**
 - Giao diện tự động cập nhật khi người chơi thực hiện hành động.
 - Sử dụng messagebox để hiển thị thông báo kết quả thắng/thua.

3.3. Luồng xử lý thuật toán chính

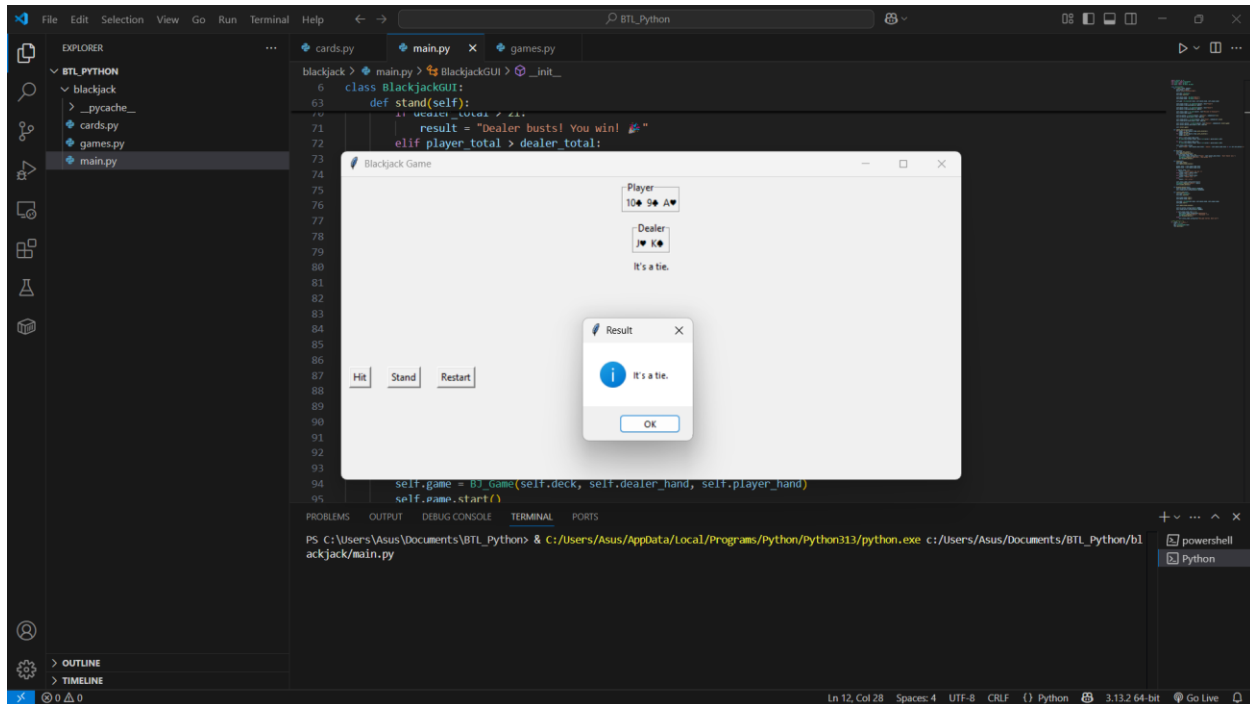
- **Khi bắt đầu hoặc nhấn nút “Restart”:**
 - Bộ bài được tạo mới và xáo lại.
 - Mỗi bên (người chơi và nhà cái) nhận hai lá bài.
 - Giao diện cập nhật để hiển thị lá bài và điểm số.



- **Khi người chơi nhấn “Hit”:**
 - Một lá bài mới được thêm vào tay bài của người chơi.
 - Nếu tổng điểm vượt quá 21 → người chơi thua (bị “Bust”).



- **Khi người chơi nhấn “Stand”:**
 - Nhà cái tự rút bài cho đến khi đạt ít nhất 17 điểm.
 - So sánh điểm để xác định kết quả trận đấu.



3.4. Giao diện đồ họa (GUI)

- Sử dụng thư viện tkinter để xây dựng giao diện trực quan.
- Hiện thị tên người chơi, lá bài, điểm số theo thời gian thực.
- Ba nút điều khiển đơn giản: **Hit**, **Stand**, **Restart**.
- Giao diện thân thiện, dễ sử dụng, cập nhật liên tục sau mỗi hành động.

3.5. Ưu điểm thiết kế

- **Tách biệt rõ ràng giữa giao diện và logic trò chơi.**
- **Dễ mở rộng:** có thể thêm chức năng đặt cược, thống kê, nhiều người chơi.
- **Hiệu quả và dễ kiểm soát:** Sử dụng OOP giúp mã nguồn ngắn gọn và dễ bảo trì.

Tóm tắt chương

Chương này trình bày quá trình thiết kế và xây dựng chương trình Blackjack với giao diện đồ họa sử dụng thư viện tkinter. Hệ thống được chia thành ba file chính: cards.py xử lý tạo lá bài và bộ bài; games.py quản lý điểm số và logic của trò chơi; và main.py xây dựng giao diện người dùng với các nút điều khiển như “Hit”, “Stand” và “Restart”. Chương trình được lập trình hướng đối tượng với các lớp như Card, BJ_Deck, BJ_Hand, và BJ_Game, giúp tổ chức mã rõ ràng, dễ mở rộng. Giao diện được cập nhật theo thời gian thực, người chơi có thể rút bài, dừng rút, và hệ thống sẽ tự động xử lý thắng/thua theo đúng luật Blackjack cơ bản.

CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT LUẬN

4.1. Đánh giá kết quả đạt được

Sau quá trình tìm hiểu và xây dựng, chương trình trò chơi Blackjack đã được hoàn thiện với đầy đủ các chức năng cơ bản như chia bài (Deal), rút bài (Hit), dừng rút (Stand), tính điểm và xác định kết quả thắng/thua. Hệ thống sử dụng mô hình lập trình hướng đối tượng rõ ràng với các lớp như Card, BJ_Deck, BJ_Hand, và BJ_Game, giúp quản lý logic trò chơi hiệu quả.

Giao diện được xây dựng bằng thư viện tkinter với các nút điều khiển trực quan, dễ sử dụng, đảm bảo tương tác tốt với người chơi. Các tình huống thường gặp trong Blackjack như “bust” (quá 21 điểm), “Blackjack” (21 điểm ngay từ đầu) hay so sánh điểm giữa người chơi và máy được xử lý hợp lý, giúp trò chơi vận hành mượt mà và đúng luật.

4.2. Hạn chế

Mặc dù chương trình đã đáp ứng các yêu cầu cơ bản, nhưng vẫn còn một số điểm hạn chế cần khắc phục:

- Giao diện còn đơn giản, chưa có hình ảnh lá bài minh họa nên chưa tạo được trải nghiệm sinh động cho người chơi.
- Trò chơi hiện tại chỉ hỗ trợ chơi một ván, không có tính năng tiếp tục nhiều ván chơi liên tiếp hay lưu lại kết quả điểm số.
- Thiếu các yếu tố nâng cao như âm thanh, hiệu ứng hình ảnh, hoặc chế độ chơi nhiều người.

4.3. Hướng phát triển

Để nâng cấp và hoàn thiện trò chơi hơn nữa, có thể mở rộng theo các hướng sau:

- **Thêm hình ảnh lá bài:** Sử dụng các file ảnh để hiển thị từng quân bài thay vì chỉ dùng ký hiệu chữ, giúp tăng tính trực quan và hấp dẫn.
- **Hỗ trợ nhiều người chơi:** Cho phép nhiều người chơi cùng tham gia, thay vì chỉ đối đầu với máy.
- **Thêm hiệu ứng âm thanh, nhạc nền:** Tạo cảm giác hứng thú hơn khi chơi.
- **Lưu điểm số, lịch sử ván chơi:** Giúp người chơi theo dõi tiến trình và so sánh kết quả qua các lần chơi.

4.4. Kết luận

Thông qua bài tập lớn này, em đã vận dụng hiệu quả kiến thức lập trình hướng đối tượng và thiết kế giao diện với tkinter vào một ứng dụng thực tế là trò chơi Blackjack. Quá trình thực hiện không chỉ giúp em củng cố kỹ năng lập trình Python mà còn rèn luyện khả năng phân tích, thiết kế và triển khai phần mềm. Đây là một trải nghiệm học tập bổ ích và thú vị, tạo tiền đề tốt cho các dự án phần mềm lớn hơn trong tương lai.

Tóm tắt chương

Chương này tổng kết quá trình xây dựng trò chơi Blackjack với giao diện đồ họa, trong đó các chức năng cơ bản như chia bài, rút bài, dừng rút và xác định kết quả đã được hoàn thiện và hoạt động đúng luật. Chương trình vận dụng hiệu quả lập trình hướng đối tượng và thư viện tkinter, tạo ra giao diện dễ sử dụng. Tuy nhiên, trò chơi còn một số hạn chế như chưa có hình ảnh lá bài, chưa hỗ trợ chơi nhiều ván hoặc lưu điểm số. Trong tương lai, chương trình có thể được phát triển thêm với các tính năng như hiển thị ảnh lá bài, chơi nhiều người, thêm hiệu ứng âm thanh và lưu lịch sử chơi. Đây là một bài tập giúp rèn luyện tốt kỹ năng lập trình và thiết kế ứng dụng thực tế.

KẾT LUẬN

Qua quá trình thực hiện bài tập lớn, em đã xây dựng thành công một trò chơi Blackjack đơn giản có giao diện đồ họa sử dụng thư viện tkinter trong Python. Từ việc tìm hiểu luật chơi cơ bản của Blackjack (Chương 2), em đã áp dụng lập trình hướng đối tượng để thiết kế các lớp như Card, BJ_Deck, BJ_Hand, và BJ_Game, tổ chức chương trình một cách rõ ràng và có tính mở rộng cao (Chương 3). Giao diện người dùng được xây dựng thân thiện, cho phép người chơi tương tác qua các nút “Deal”, “Hit”, “Stand” và hệ thống tự động tính điểm, xác định thắng thua theo luật.

Đề tài đã giúp em rèn luyện và củng cố các kiến thức quan trọng như: tư duy thiết kế chương trình, xử lý sự kiện GUI, cũng như vận dụng hiệu quả các cấu trúc dữ liệu trong Python (Chương 1 & 2). Mặc dù giao diện còn đơn giản và thiếu một số tính năng nâng cao như hiển thị ảnh bài hoặc lưu điểm, nhưng sản phẩm đã đạt được mục tiêu cơ bản đề ra. Trong tương lai, em mong muốn tiếp tục cải tiến trò chơi bằng cách bổ sung hình ảnh, âm thanh và mở rộng chức năng chơi nhiều người (Chương 4).

Bài tập lớn này không chỉ giúp em nâng cao kỹ năng lập trình mà còn tạo cơ hội áp dụng kiến thức vào một sản phẩm cụ thể, từ đó tăng khả năng tư duy logic và phát triển phần mềm một cách bài bản. Đây là một trải nghiệm học tập hữu ích và đầy hứng thú.

TÀI LIỆU THAM KHẢO

Giáo trình Python – Chương 9: Lập trình trò chơi đơn giản

Nơi cung cấp nền tảng kiến thức về thiết kế trò chơi theo hướng đối tượng và mô hình hóa logic game Blackjack.

Tài liệu chính thức Python (<https://docs.python.org/3/>)

Tham khảo cú pháp, cách sử dụng các cấu trúc dữ liệu, hàm built-in và nguyên lý hoạt động của Python.

Tài liệu thư viện tkinter – Python GUI Programming

Nguồn: <https://docs.python.org/3/library/tk.html>

Giải thích chi tiết cách sử dụng các widget như Tk, Label, Button, Frame và quản lý sự kiện trong tkinter.

Trang W3Schools – Python OOP và GUI với tkinter

Nguồn: <https://www.w3schools.com/python/>

Tài nguyên học lập trình Python và tkinter theo từng bước cụ thể, dễ hiểu.

Stack Overflow

Trang hỗ trợ kỹ thuật, tham khảo cách xử lý lỗi, tối ưu logic khi gặp vấn đề trong quá trình viết và chạy chương trình.

YouTube – Hướng dẫn xây dựng trò chơi Blackjack bằng Python

Một số video hướng dẫn từng bước xây dựng Blackjack và thiết kế GUI đơn giản, giúp hình dung rõ hơn về quy trình lập trình game.

ChatGPT – OpenAI

Hỗ trợ giải thích logic, gợi ý cấu trúc chương trình, sửa lỗi nội dung báo cáo bài tập lớn một cách hệ thống.