**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**



**KHOA ĐIỆN TỬ**

**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**

✯✯✯

Ảnh có chứa biểu tượng, Nhãn hiệu, phù hiệu, vòng tròn

Mô tả được tạo tự động

**BÀI TẬP KẾT THÚC MÔN HỌC**

**MÔN HỌC**

**LẬP TRÌNH PYTHON**

|  |  |  |
| --- | --- | --- |
| **d** | **:** | **Nguyễn Văn Huy** |
| **Họ tên** | **:** | **Dương Thị Ly** |
| **MSV** | **:** | **K225480106045** |
| **Lớp** | **:** | **K58KTP** |
| **Link github** | **:** | **https://github.com/Duonglly/BTKTM\_python\_Blackjack.git** |
| **Link ytb** | **:** | **https://youtu.be/oUR67ivWmg8** |

**Thái Nguyên 2025**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC KTCN |  |
| KHOA ĐIỆN TỬ  BỘ MÔN CÔNG NGHỆ THÔNG TIN | Độc lập - Tự do - Hạnh phúc  --------o0o--------- |

**BÀI TẬP KẾT THÚC MÔN HỌC**

MÔN HỌC: LẬP TRÌNH PYTHON

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Họ tên sinh viên: Dương Thị ly

Msv: K225480106045

Lớp : K58KTP Ngành : Kỹ thuật phần mềm

Giáo viên hướng dẫn: Nguyễn Văn Huy

*Ngày giao đề:20/5/2025*

*Ngày hoàn thành:10/6/2025*

1. Tên đề tài :

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

2. Yêu cầu của đề tài:

* Import module cards, games.
* Class GUI: Frame người chơi và dealer.
* Button “Hit” và “Stand” gọi phương thức game.
* Cập nhật kết quả trên Label/Messagebox.

|  |  |
| --- | --- |
| TRƯỞNG BỘ MÔN  (Ký và ghi rõ họ tên) | GIÁO VIÊN HƯỚNG DẪN  (Ký và ghi rõ họ tên) |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Thái Nguyên, ngày 10 tháng 06 năm 2025.*

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên)*

**MỤC LỤC**

[LỜI NÓI ĐẦU......................................................................................... 7](#_Toc200051012)

[CHUƠNG I. GIỚI THIỆU ĐỀ TÀI..........................................................8](#_Toc200051015)

1.1.[Mục tiêu của đề tài..........................................................................8](#_Toc200051016)

1.2.[Nội dung, phạm vi thực hiện...........................................................8](#_Toc200051017)

[CHƯƠNG 2. PHƯƠNG PHÁP THỰC HIỆN........................................10](#_Toc200051036)

[2.1. Phân tích yêu cầu.........................................................................10](#_Toc200051037)

[2.1.1. Yêu cầu chức năng.............................................................. 10](#_Toc200051038)

[2.1.2. Luồng xử lý và tương tác người dùng................................ 10](#_Toc200051039)

[2.1.3.Sơ đồ chức năng.................................................................. 11](#_Toc200051040)

[CHƯƠNG 3: THIẾT KẾ.........................................................................13](#_Toc200051041)

[3.1. Kiến trúc hệ thống........................................................................13](#_Toc200051042)

[3.2. Thiết kế giao diện người chơi......................................................14](#_Toc200051043)

[3.3. Thiết kế lớp và xử lý logic game.................................................15](#_Toc200051048)

[CHƯƠNG 4:CÀI ĐẶT VÀ THỰC HIỆN THIẾT KẾ GAME.............. 17](#_Toc200051049)

[4.1. Cài đặt môi trường và tài nguyên game...........................................17](#_Toc200051050)

[4.1.1. Môi trường phát triển..........................................................17](#_Toc200051051)

[4.1.2. Tài nguyên game sử dụng...................................................17](#_Toc200051052)

[4.1.3. Cấu trúc thư mục dự án.......................................................18](#_Toc200051053)

[4.2. Cài đặt chức năng chính...............................................................19](#_Toc200051058)

[4.2.1. Khởi tạo và nạp tài nguyên.................................................19](#_Toc200051059)

[4.2.2. Xử lý chia bài và tính điểm.................................................20](#_Toc200051060)

[4.2.3. Vẽ giao diện và thông tin ...................................................21](#_Toc200051061)

[4.2.4. Điều khiển luồng trò chơi...................................................22](#_Toc200051062)

[4.2.5. Tổ chức dữ liệu...................................................................22](#_Toc200051063)

[4.3. Hiệu ứng âm thanh, hình ảnh và trải nghiệm người dùng............23](#_Toc200051065)

[4.3.1. Hình ảnh.............................................................................23](#_Toc200051066)

[4.3.2. Âm thanh..............................................................................23](#_Toc200051067)

[4.3.3. Trải nghiệm người dùng (UX/UI) .......................................24](#_Toc200051068)

[4.4. Toàn bộ code chương trình............................................................24](#_Toc200051069)

[CHƯƠNG 5:THỬ NGHIỆM....................................................................33](#_Toc200051070)

[5.1. Kiểm thử chức năng chính.............................................................33](#_Toc200051071)

[5.1.1. Giao diện hướng dẫn.............................................................33](#_Toc200051072)

[5.1.2. Rút bài – phím H.............................................................33](#_Toc200051073)

[5.1.3. Dừng bài – phím S................................................................34](#_Toc200051074)

[5.1.4. Thắng bài..............................................................................34](#_Toc200051075)

[5.1.5. Thua bài................................................................................35](#_Toc200051076)

[5.1.6. Hòa........................................................................................35](#_Toc200051077)

[5.1.7. Phát bài ban đầu....................................................................36](#_Toc200051078)

[5.2. Kết quả chạy thử và đánh giá.........................................................36](#_Toc200051079)

[CHƯƠNG 6:KẾT LUẬN.... .....................................................................38](#_Toc200051081)

[6.1. Tổng kết nội dung báo cáo.............................................................38](#_Toc200051082)

[6.2. Đánh giá kết quả đạt được..............................................................38](#_Toc200051083)

[6.3. Hướng phát triển trong tương lai....................................................38](#_Toc200051084)

[6.4. Kết luận chung................................................................................38](#_Toc200051085)

[TÀI LIỆU THAM KHẢO.........................................................................39](#_Toc200051086)

# **DANH MỤC HÌNH ẢNH**

[Hình 2.1.Sơ đồ chức năng 12](#_Toc200053458)

[Hình 3.1. Sơ đồ kiến trúc tổng thể 14](#_Toc200053459)

[Hình 3.2. Sơ đồ lớp 15](#_Toc200053460)

[Hình 4.1. Thư viện cài đặt pygame 17](#_Toc200053461)

[Hình 4.2. Thư mục âm thanh 18](#_Toc200053462)

[Hình 4.3. Cấu trúc thư mục dự án 18](#_Toc200053463)

[Hình 4.4. Khởi tạo và nạp tài nguyên 19](#_Toc200053464)

[Hình 4.5. Xử lý chia bài và tính điểm 20](#_Toc200053465)

[Hình 4.6. Vẽ giao diện và thông tin 21](#_Toc200053466)

[Hình 4.7. Điều khiển luồng trò chơi 22](#_Toc200053467)

[Hình 4.8. Sơ đồ quy trình chức năng chính 23](#_Toc200053468)

[Hình 5.1. Giao diện hướng dẫn 33](#_Toc200053469)

[Hình 5.2. Rút bài – phím H 33](#_Toc200053470)

[Hình 5.3. Dừng bài – phím S 34](#_Toc200053471)

[Hình 5.4. Thắng bài 34](#_Toc200053472)

[Hình 5.5. Thua bài 35](#_Toc200053473)

[Hình 5.6. Hòa 35](#_Toc200053474)

[Hình 5.7. Phát bài ban đầu 36](#_Toc200053475)

# **LỜI NÓI ĐẦU**

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, ngành công nghiệp trò chơi điện tử không chỉ là lĩnh vực giải trí đơn thuần mà còn là một ngành công nghiệp mang lại giá trị kinh tế lớn, đồng thời góp phần thúc đẩy sự sáng tạo và phát triển kỹ năng lập trình của sinh viên công nghệ thông tin

Blackjack là một trò chơi bài phổ biến, đòi hỏi người chơi phải tính toán điểm số và ra quyết định hợp lý. Thông qua việc xây dựng trò chơi này, em có cơ hội áp dụng các kiến thức đã học về lập trình hướng đối tượng, xử lý giao diện đồ họa với thư viện Pygame, thao tác với hình ảnh, âm thanh và xây dựng logic xử lý tình huống trong trò chơi.

Bài tập giúp em hiểu rõ hơn về quy trình xây dựng một ứng dụng hoàn chỉnh – từ phân tích yêu cầu, thiết kế giao diện, đến lập trình và thử nghiệm sản phẩm. Đồng thời, em cũng rèn luyện được kỹ năng tự học, giải quyết vấn đề và tư duy logic.

Em xin chân thành cảm ơn thầy Nguyễn Văn Huy giảng viên hướng dẫn, đã tận tình hỗ trợ và tạo điều kiện để em hoàn thành bài tập này. Em cũng xin cảm ơn các thầy cô trong bộ môn đã truyền đạt những kiến thức quý báu trong suốt học kỳ vừa qua.

Mặc dù đã cố gắng hoàn thành bài tập một cách tốt nhất, nhưng do thời gian có hạn và kinh nghiệm thực tế còn hạn chế, bài làm không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý của cô và quý thầy cô để hoàn thiện hơn trong các dự án sau này.

# **CHUƠNG I. GIỚI THIỆU ĐỀ TÀI**

1.1.Mục tiêu của đề tài

* Hiểu và áp dụng kiến thức lập trình Python vào việc xây dựng một trò chơi có luật chơi cụ thể, logic rõ ràng và tương tác với người dùng.
* Làm quen với thư viện Pygame, sử dụng để xử lý đồ họa, âm thanh và sự kiện trong trò chơi.
* Thiết kế và triển khai một hệ thống trò chơi mô phỏng Blackjack, trong đó người chơi có thể tương tác bằng bàn phím hoặc chuột, bao gồm các chức năng như chia bài, tính điểm, xử lý thắng – thua, phát nhạc nền và hiệu ứng âm thanh.
* Tăng cường tư duy lập trình hướng đối tượng, thông qua việc xây dựng các lớp (class) như bài, bộ bài, người chơi, nhà cái, v.v.
* Phát triển kỹ năng phân tích, thiết kế và triển khai phần mềm, thông qua quy trình hoàn chỉnh từ phân tích yêu cầu, thiết kế hệ thống, viết mã nguồn, kiểm thử và đánh giá.
* Tạo ra một sản phẩm phần mềm có tính trực quan, dễ sử dụng và giải trí, có thể mở rộng trong tương lai như thêm chức năng đặt cược, nhiều người chơi hoặc trí tuệ nhân tạo.

## 1.2.Nội dung, phạm vi thực hiện

1. Nội dung thực hiện

* Phân tích luật chơi Blackjack và xây dựng logic xử lý điểm số, luật rút bài, thắng – thua.
* Thiết kế giao diện đồ họa trò chơi bằng thư viện Pygame, bao gồm nền bàn chơi, các lá bài, nút chức năng và hiển thị điểm số.
* Xây dựng chức năng điều khiển trò chơi thông qua bàn phím (phím Hit – Rút bài, Stand – Dừng) và chuột (chọn nút PLAY, PLAY AGAIN).
* Tích hợp âm thanh gồm nhạc nền và các hiệu ứng như chia bài, thắng hoặc thua.
* Xử lý các trạng thái của trò chơi: đang chơi, kết thúc, chờ người chơi bắt đầu lại.
* Tổ chức mã nguồn rõ ràng, dễ mở rộng, sử dụng mô hình hướng đối tượng kết hợp xử lý sự kiện Pygame.

1. Phạm vi thực hiện

* Xây dựng phiên bản đơn giản của trò chơi Blackjack với một người chơi và một nhà cái (máy).
* Trò chơi chạy trên môi trường máy tính cá nhân, không hỗ trợ nhiều người chơi hoặc chơi qua mạng.
* Không bao gồm các tính năng nâng cao như cá cược, lưu điểm cao, đăng nhập người dùng, AI nhà cái chuyên sâu.
* Tập trung vào logic trò chơi, thiết kế đồ họa cơ bản và khả năng hoàn thiện trong phạm vi môn học.

## 1.3.Kết quả cần đạt được

* Xây dựng thành công một trò chơi Blackjack hoàn chỉnh hoạt động trên máy tính cá nhân, có thể chơi giữa người dùng và máy (dealer).
* Hiểu rõ và áp dụng đúng luật chơi Blackjack vào hệ thống xử lý điểm, rút bài, xử lý thắng/thua.
* Thiết kế giao diện trực quan, dễ sử dụng, bao gồm hình ảnh nền, các lá bài, vùng hiển thị điểm số và nút điều khiển.
* Tích hợp hiệu ứng âm thanh cho các hành động như rút bài, thắng, thua, và nhạc nền xuyên suốt trò chơi.
* Tổ chức mã nguồn mạch lạc, dễ mở rộng, sử dụng lập trình hướng đối tượng và thư viện Pygame một cách hợp lý.

# **CHƯƠNG 2. PHƯƠNG PHÁP THỰC HIỆN**

## 2.1. Phân tích yêu cầu

### 2.1.1. Yêu cầu chức năng

* Người chơi có thể bắt đầu trò chơi từ màn hình hướng dẫn với nút “Play Now”.
* Hiển thị hai lá bài đầu tiên cho người chơi và nhà cái (dealer), trong đó 1 lá bài của dealer bị úp.
* Cho phép người chơi thực hiện hành động:
* Hit: rút thêm 1 lá bài.
* Stand: dừng rút bài và chuyển lượt cho dealer.
* Dealer tự động rút bài đến khi đạt ít nhất 17 điểm.
* Tính toán và hiển thị điểm số của người chơi và dealer.
* Xác định người chiến thắng dựa trên điểm số, theo đúng luật Blackjack:
* 21 điểm là thắng tuyệt đối.
* Quá 21 điểm bị xử thua (bust).
* Điểm cao hơn đối phương sẽ thắng.
* Hiển thị giao diện kết thúc trò chơi, gồm:
* Kết quả trận đấu (Win/Lose/Draw).
* Các lá bài của cả hai bên.
* Nút “Play Again” để chơi lại.
* Tích hợp hiệu ứng âm thanh cho các sự kiện chính như rút bài, thắng, thua, nhạc nền.

### 2.1.2. Luồng xử lý và tương tác người dùng

#### Màn hình khởi động / hướng dẫn:

* Khi mở trò chơi, người dùng sẽ thấy màn hình chào mừng với phần hướng dẫn luật chơi cơ bản.
* Người dùng nhấn vào nút "PLAY NOW" để bắt đầu ván bài.

#### Giai đoạn chia bài:

* Hệ thống sẽ tự động chia 2 lá bài cho người chơi và 2 lá bài cho dealer.
* Một lá bài của dealer sẽ được úp xuống, chưa cho người chơi thấy.
* Điểm số của người chơi được hiển thị ngay sau khi chia bài.

#### Lượt chơi của người dùng:

* Người dùng có thể:
* Nhấn phím "H" (Hit) để rút thêm một lá bài.
* Nhấn phím "S" (Stand) để kết thúc lượt và chuyển sang lượt của dealer.
* Sau mỗi hành động "Hit", hệ thống:
* Thêm một lá bài mới vào bộ bài của người chơi.
* Cập nhật và hiển thị điểm mới.
* Phát âm thanh rút bài (nếu có).
* Nếu điểm vượt quá 21:
* Người chơi bị xử thua (bust).
* Trò chơi chuyển sang màn hình kết thúc.

#### Lượt chơi của dealer (máy):

* Khi người chơi chọn "Stand", hệ thống:
* Tự động để dealer rút bài đến khi đủ 17 điểm trở lên hoặc bị bust.
* Cập nhật điểm số liên tục cho dealer.
* Sau đó, hệ thống so sánh điểm số giữa người chơi và dealer để xác định kết quả.

#### Màn hình kết thúc trò chơi:

* Hiển thị:
* Tất cả các lá bài của cả hai bên.
* Điểm số của người chơi và dealer.
* Kết quả: Thắng (Win), Thua (Lose), Hòa (Tie).
* Kèm hiệu ứng âm thanh thắng/thua tương ứng.
* Cung cấp nút “PLAY AGAIN” để người chơi bắt đầu lại một ván mới.

#### Thoát trò chơi:

* Người chơi có thể thoát trò chơi bất kỳ lúc nào bằng cách nhấn nút [X] trên cửa sổ hoặc tổ hợp phím Alt + F4.

### 2.1.3.Sơ đồ chức năng

#### **Mô tả tổng quan**

Sơ đồ chức năng mô tả các tác vụ chính mà người chơi (user) có thể thực hiện khi tương tác với hệ thống trò chơi Blackjack. Trong hệ thống này, người chơi là tác nhân duy nhất, tương tác với các chức năng chính như: bắt đầu trò chơi, rút bài, dừng bài, xem kết quả và chơi lại.

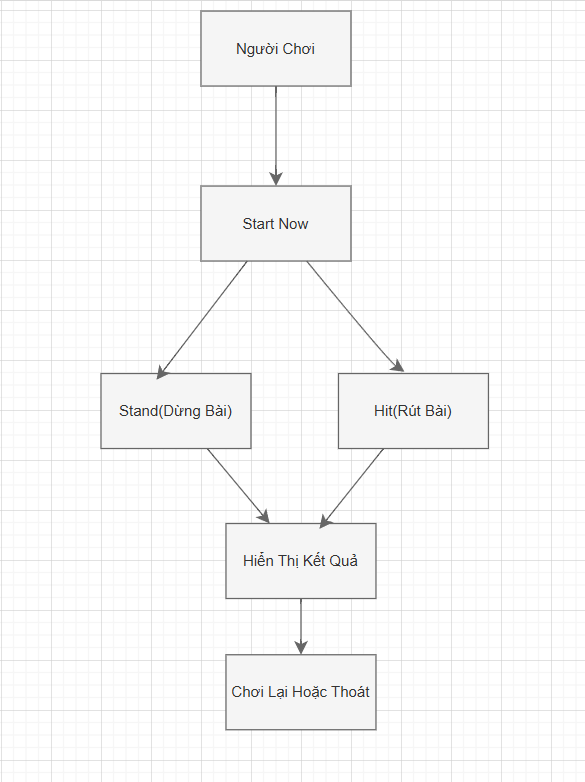
#### **Các tác nhân (Actors)**

* Người chơi (Player): Tác nhân chính, điều khiển toàn bộ trò chơi thông qua bàn phím và chuột.

#### **Các chức năng chính (Use Cases)**

* Xem hướng dẫn trò chơi: Người chơi được cung cấp thông tin luật chơi và cách thao tác trước khi bắt đầu.
* Bắt đầu trò chơi: Người chơi nhấn "PLAY NOW" để khởi động một ván bài mới.
* Rút bài (Hit): Người chơi nhấn phím H để yêu cầu thêm một lá bài.
* Dừng bài (Stand): Người chơi nhấn phím S để kết thúc lượt chơi của mình, chuyển sang lượt của Dealer.
* Tự động chia bài cho Dealer: Hệ thống sẽ tự xử lý các lượt rút bài của Dealer.
* Hiển thị kết quả: Trò chơi hiển thị điểm số và kết quả thắng/thua/hòa.
* Chơi lại (Play Again): Cho phép người chơi bắt đầu lại từ đầu sau khi ván trước kết thúc.
* Thoát trò chơi: Người chơi có thể thoát ứng dụng bất kỳ lúc nào.

#### **Sơ đồ minh họa**

****

Hình 2.1.Sơ đồ chức năng

# 

# **CHƯƠNG 3: THIẾT KẾ**

## 3.1. Kiến trúc hệ thống

Hệ thống trò chơi Blackjack được xây dựng trên nền tảng Python sử dụng thư viện **Pygame** để xử lý đồ họa và âm thanh. Kiến trúc hệ thống được thiết kế theo mô hình **một lớp chính điều khiển luồng chương trình (Main Loop),** trong đó các chức năng được phân chia rõ ràng theo từng nhiệm vụ, bao gồm:

#### **Tầng giao diện người dùng (User Interface)**

* Hiển thị cửa sổ trò chơi, nền sòng bài, các lá bài và văn bản hướng dẫn.
* Xử lý tương tác người dùng thông qua bàn phím (phím H, S) và chuột (nhấp nút “Play Now”, “Play Again”).
* Cập nhật trạng thái màn hình theo từng hành động của người chơi và dealer.

#### **Tầng xử lý trò chơi (Game Logic)**

* Thực hiện các thao tác chính của trò chơi như:
* Chia bài cho người chơi và dealer.
* Tính điểm số từ bộ bài của người chơi và dealer.
* Kiểm tra điều kiện thắng/thua/hòa.
* Quản lý các trạng thái của trò chơi (đang chơi, kết thúc, chơi lại...).

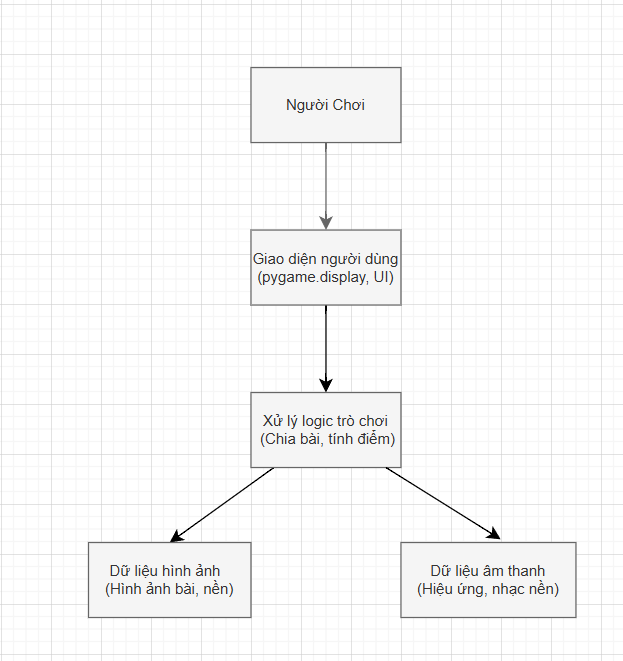
#### **Tầng dữ liệu (Assets – Âm thanh và hình ảnh)**

* Gồm:
* Hình ảnh của các lá bài (52 lá chia theo 4 chất).
* Hình nền sòng bài, mặt sau lá bài.
* Các hiệu ứng âm thanh: chia bài, thắng cuộc, thua cuộc, nhạc nền.
* Các dữ liệu này được tải và xử lý khi khởi động trò chơi.

#### **Tầng âm thanh (Sound Layer)**

* Tải và điều khiển các hiệu ứng âm thanh bằng pygame.mixer.Sound.
* Chơi nhạc nền theo vòng lặp (pygame.mixer.music.play(-1))

1. **Sơ đồ kiến trúc tổng thể**



Hình 3.1. Sơ đồ kiến trúc tổng thể

## 3.2. Thiết kế giao diện người chơi

### Màn hình chính (Main Menu)

* Chức năng:
* Bắt đầu chơi mới (New Game)
* Xem hướng dẫn (Instructions)
* Thoát game (Exit)
* Thiết kế:
* Nút chức năng lớn, dễ nhấn
* Logo game và nền giao diện bắt mắt

### Màn hình chơi game (Gameplay Screen)

* Khu vực người chơi:
* Hiển thị các lá bài đã rút
* Hiển thị tổng điểm hiện tại
* Khu vực đặt cược (nếu có)
* Khu vực nhà cái (Dealer):
* Hiển thị các lá bài (một lá úp, một lá ngửa)
* Tự động lật bài khi đến lượt
* Các nút chức năng:
* Hit – Rút thêm bài
* Stand – Dừng rút bài
* Restart – Chơi lại ván mới
* Thông báo kết quả:
* Hiển thị kết quả thắng/thua/hòa sau mỗi ván
* Âm thanh thông báo tương ứng với trạng thái

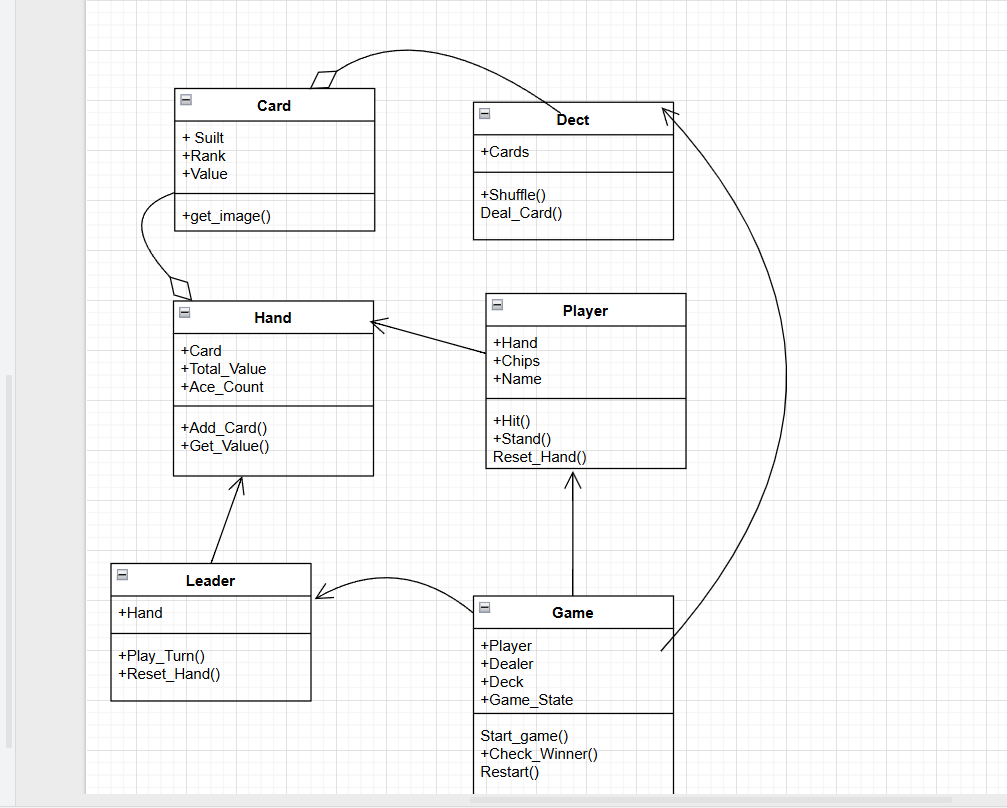
### Giao diện trực quan và hiệu ứng

* Sử dụng ảnh nền bàn chơi (table background) tạo cảm giác như sòng bài thật.
* Lá bài được hiển thị bằng hình ảnh trực quan với hiệu ứng lật bài.
* Âm thanh đi kèm khi rút bài, chiến thắng, thua cuộc tăng sự hấp dẫn.

### Tương thích và kích thước

* Giao diện thiết kế phù hợp với độ phân giải 1280x720 trở lên.
* Tương thích với màn hình máy tính chuẩn, điều khiển bằng chuột.

## 3.3. Thiết kế lớp và xử lý logic game



Hình 3.2. Sơ đồ lớp

#### **Lớp** Card

* Đại diện cho một lá bài cụ thể.
* Thuộc tính:
  + suit: Chất (♠, ♥, ♦, ♣)
  + rank: Hạng (A, 2–10, J, Q, K)
  + value: Giá trị điểm
* Phương thức:
  + get\_image(): Lấy hình ảnh tương ứng của lá bài

#### **Lớp** Deck

* + Quản lý bộ bài gồm 52 lá.
  + Phương thức:
    - shuffle(): Xáo bài
    - deal\_card(): Rút một lá bài

#### **Lớp** Hand

* + Đại diện cho các lá bài trên tay người chơi hoặc dealer.
  + Quản lý điểm tổng và số lượng lá bài A (để xử lý logic 1 hoặc 11 điểm).
  + Phương thức:
  + add\_card(card): Thêm lá bài vào tay
  + get\_value(): Tính tổng điểm tay bài

#### **Lớp** Player

* + Đại diện người chơi.
  + Quản lý hand, số tiền cá cược (nếu có).
  + Phương thức:
    - hit(): Rút bài
    - stand(): Ngừng rút
    - reset\_hand(): Dọn bài cho ván mới

#### **Lớp** Dealer

* + Tự động rút bài theo luật nhà cái.
  + Phương thức:
    - play\_turn(): Logic tự động rút khi điểm < 17
    - reset\_hand(): Dọn bài

#### **Lớp** Game

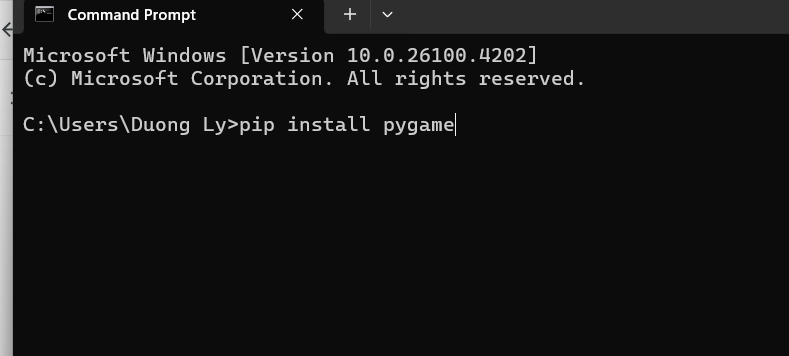
* + Quản lý toàn bộ quá trình chơi.
  + Giao tiếp giữa player – dealer – deck.
  + Phương thức:
    - start\_game(): Khởi tạo ván chơi mới
    - check\_winner(): So sánh điểm và xác định thắng thua
    - restart(): Reset trạng thái game

# **CHƯƠNG 4:CÀI ĐẶT VÀ THỰC HIỆN THIẾT KẾ GAME**

## 4.1. Cài đặt môi trường và tài nguyên game

### 4.1.1. Môi trường phát triển

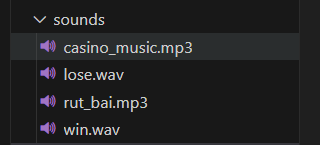
* Ngôn ngữ lập trình: Python 3.10 trở lên
* Thư viện đồ họa: pygame – hỗ trợ xử lý đồ họa, âm thanh và sự kiện bàn phím/chuột.
* Công cụ lập trình:
* Trình soạn thảo mã nguồn: VS Code
* Trình quản lý thư viện: pip
* Cài đặt pygame:



Hình 4.1. Thư viện cài đặt pygame

### 4.1.2. Tài nguyên game sử dụng

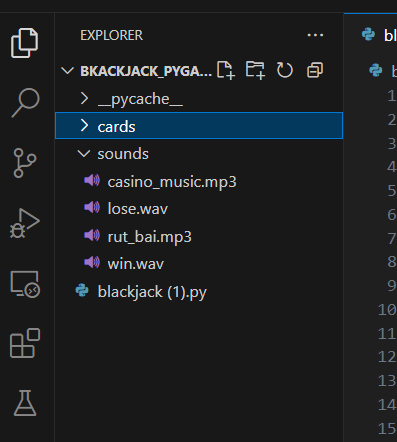
* Hình ảnh lá bài:
  + Bộ ảnh 52 lá bài tiêu chuẩn định dạng .png, gồm các giá trị từ A → K và 4 chất (♥, ♣, ♦, ♠).
  + Ảnh mặt sau lá bài (back.png).
* Âm thanh:



Hình 4.2. Thư mục âm thanh

* + Âm thanh rút bài:rut\_bai.mp3
  + Âm thanh thắng/thua: win.wav, lose.wav
  + Nhạc nền lặp: casino\_music.mp3

### 4.1.3. Cấu trúc thư mục dự án

1. 

Hình 4.3. Cấu trúc thư mục dự án

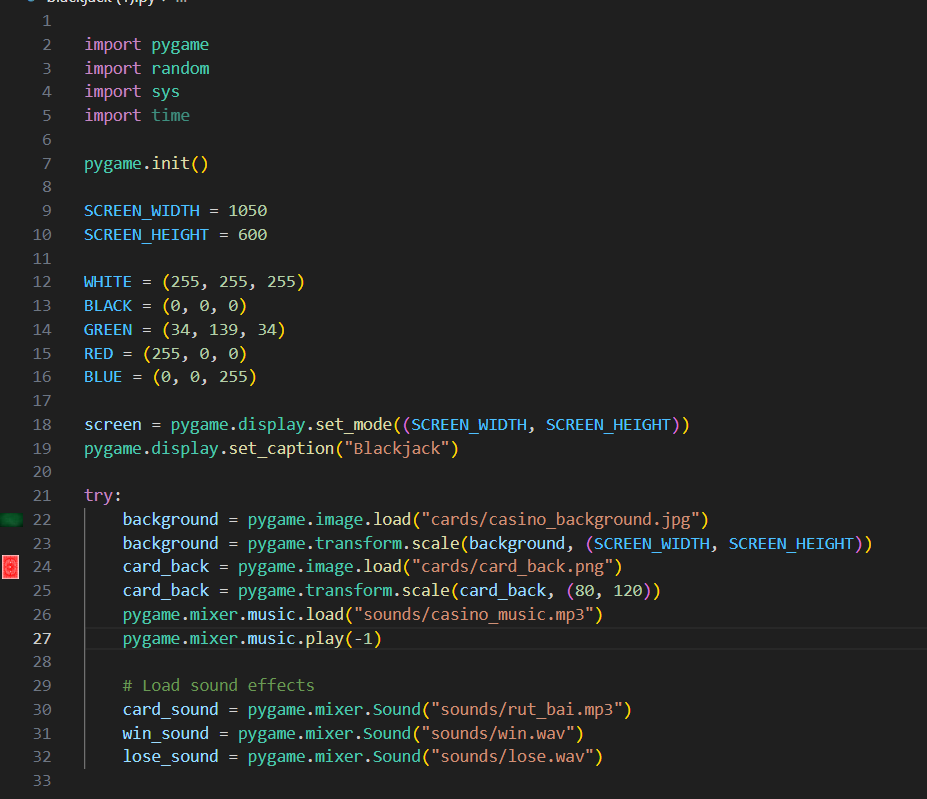
-Card: Chứa hình ảnh các lá bài

-Sounds: Chứa âm thanh

-Blackjack(1).py: Chứa toàn bộ code tạo nên trò chơi

## 4.2. Cài đặt chức năng chính

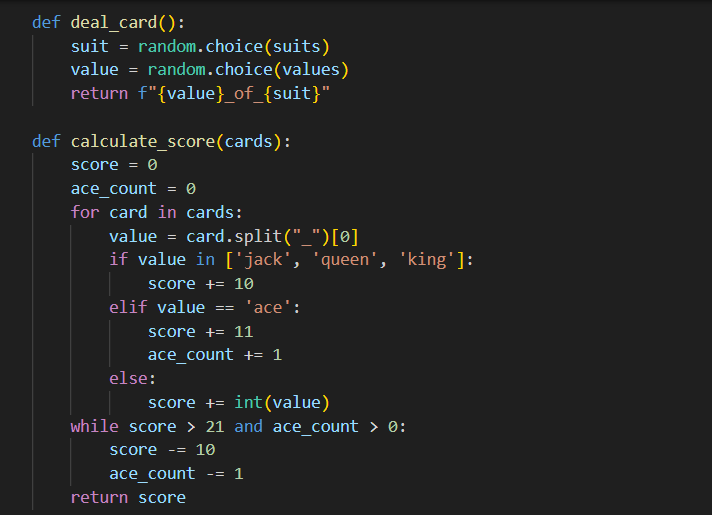
### 4.2.1. Khởi tạo và nạp tài nguyên



Hình 4.4. Khởi tạo và nạp tài nguyên

* Khởi động thư viện pygame bằng pygame.init().
* Thiết lập kích thước cửa sổ game, tiêu đề và màu sắc cơ bản.
* Tải hình nền, hình ảnh các lá bài (từ thư mục cards/) và các hiệu ứng âm thanh (từ thư mục sounds/).
* Khởi động nhạc nền (casino\_music.mp3) và chuẩn bị âm thanh cho các hành động như rút bài (rut\_bai.mp3), thắng (win.wav) và thua (lose.wav).

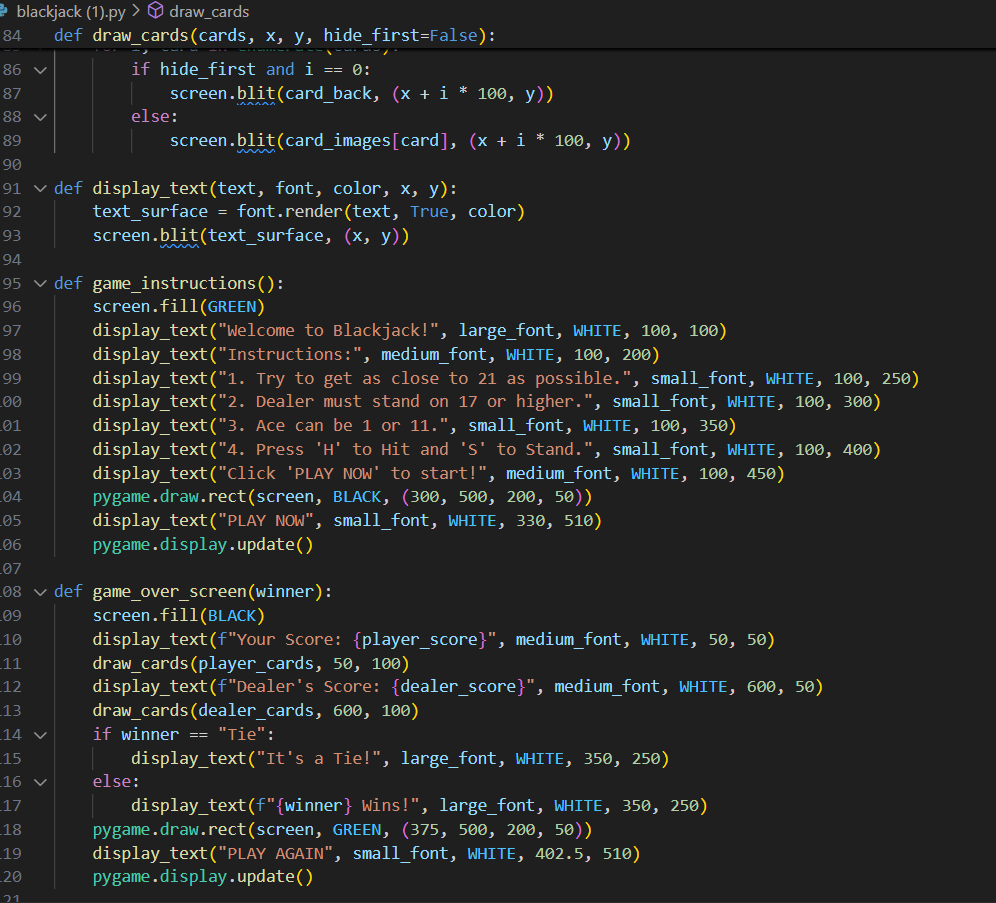
### 4.2.2. Xử lý chia bài và tính điểm



Hình 4.5. Xử lý chia bài và tính điểm

* Hàm deal\_card() chọn ngẫu nhiên 1 lá bài từ 52 lá.
* Hàm calculate\_score(cards) tính tổng điểm các lá bài đang có. Lá Át (Ace) được xử lý đặc biệt để có thể tính là 1 hoặc 11 tùy vào tổng điểm hiện tại nhằm tránh bị “quá 21”.

### 4.2.3. Vẽ giao diện và thông tin



Hình 4.6. Vẽ giao diện và thông tin

* Hàm draw\_cards(cards, x, y, hide\_first=False) dùng để vẽ các lá bài lên màn hình.
* Hàm display\_text(text, font, color, x, y) hiển thị văn bản như điểm số, hướng dẫn hoặc thông báo kết thúc trận.
* Hàm game\_instructions() hiển thị màn hình hướng dẫn đầu game với nút "PLAY NOW".
* Hàm game\_over\_screen(winner) hiển thị kết quả khi kết thúc ván chơi và cung cấp nút "PLAY AGAIN".

### 4.2.4. Điều khiển luồng trò chơi

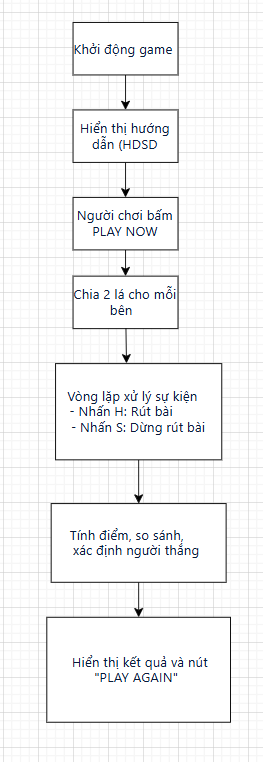
|  |  |
| --- | --- |
| **Hình 4.7. Điều khiển luồng trò chơi** |  |

* Hàm main() điều khiển toàn bộ trò chơi:
* Bắt đầu bằng màn hình hướng dẫn.
* Sau khi người chơi chọn “PLAY NOW”, hệ thống sẽ gọi reset\_game() để thiết lập lại dữ liệu.
* Trong vòng lặp chính, người chơi có thể nhấn:
* H để rút bài (Hit).
* S để dừng rút (Stand), lúc này máy sẽ thực hiện lượt rút của dealer theo luật.
* Trạng thái thắng/thua được xác định và hiển thị sau mỗi lượt chơi.

### 4.2.5. Tổ chức dữ liệu

* Các thông tin như danh sách lá bài (player\_cards, dealer\_cards), điểm số, trạng thái game (game\_over, winner) được lưu trữ dưới dạng biến toàn cục.

### 4.2.6. Sơ đồ quy trình chức năng chính



Hình 4.8. Sơ đồ quy trình chức năng chính

## 4.3. Hiệu ứng âm thanh, hình ảnh và trải nghiệm người dùng

### 4.3.1. Hình ảnh

* Hình nền: Sử dụng hình ảnh nền casino (casino\_background.jpg) tạo cảm giác chân thực như đang chơi tại sòng bạc thực tế.
* Lá bài: Toàn bộ 52 lá bài được sử dụng với hình ảnh chi tiết, chuẩn xác, được resize lại với kích thước đồng đều (80 x 120) pixel giúp hiển thị rõ ràng trên mọi độ phân giải.
* Mặt sau lá bài (card\_back.png) được sử dụng cho dealer khi ẩn bài, tạo cảm giác hồi hộp và chuyên nghiệp.
* Giao diện nút bấm (PLAY NOW, PLAY AGAIN) được hiển thị nổi bật và rõ ràng để người chơi dễ thao tác.

### 4.3.2. Âm thanh

* Nhạc nền: Tệp casino\_music.mp3 được phát liên tục trong toàn bộ trò chơi giúp người chơi cảm thấy thư giãn và tập trung.
* Hiệu ứng âm thanh:
* rut\_bai.mp3: Phát ra khi người chơi hoặc dealer rút bài, tạo cảm giác hành động rõ ràng.
* win.wav: Phát ra khi người chơi chiến thắng.
* lose.wav: Phát ra khi người chơi thua cuộc.
* Âm thanh được tích hợp thông qua pygame.mixer và được phát đúng thời điểm tương ứng với hành động.

### 4.3.3. Trải nghiệm người dùng (UX/UI)

* Hướng dẫn rõ ràng: Màn hình đầu tiên cung cấp phần hướng dẫn dễ hiểu, giúp người chơi mới tiếp cận game một cách nhanh chóng.
* Luồng chơi mạch lạc:
* Giao diện phân biệt rõ ràng giữa bài của người chơi và dealer.
* Hiển thị điểm số tức thì khi rút bài.
* Tự động xử lý phần chơi của dealer sau khi người chơi đứng bài.
* Thông báo kết quả: Giao diện kết thúc game có hiển thị điểm số chi tiết hai bên và kết quả thắng/thua rõ ràng.
* Khả năng chơi lại: Cho phép người chơi chơi nhiều ván liên tiếp với nút "PLAY AGAIN", giúp kéo dài thời gian tương tác mà không cần khởi động lại chương trình.

## 4.4. Toàn bộ code chương trình

import pygame

import random

import sys

import time

pygame.init()

SCREEN\_WIDTH = 1050

SCREEN\_HEIGHT = 600

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

GREEN = (34, 139, 34)

RED = (255, 0, 0)

BLUE = (0, 0, 255)

screen = pygame.display.set\_mode((SCREEN\_WIDTH, SCREEN\_HEIGHT))

pygame.display.set\_caption("Blackjack")

try:

    background = pygame.image.load("cards/casino\_background.jpg")

    background = pygame.transform.scale(background, (SCREEN\_WIDTH, SCREEN\_HEIGHT))

    card\_back = pygame.image.load("cards/card\_back.png")

    card\_back = pygame.transform.scale(card\_back, (80, 120))

    pygame.mixer.music.load("sounds/casino\_music.mp3")

    pygame.mixer.music.play(-1)

    # Load sound effects

    card\_sound = pygame.mixer.Sound("sounds/rut\_bai.mp3")

    win\_sound = pygame.mixer.Sound("sounds/win.wav")

    lose\_sound = pygame.mixer.Sound("sounds/lose.wav")

except pygame.error as e:

    print(f"Error loading media: {e}")

    sys.exit()

card\_images = {}

suits = ['hearts', 'diamonds', 'clubs', 'spades']

values = ['2', '3', '4', '5', '6', '7', '8', '9', '10', 'jack', 'queen', 'king', 'ace']

for suit in suits:

    for value in values:

        card\_name = f"{value}\_of\_{suit}"

        card\_images[card\_name] = pygame.image.load(f"cards/{card\_name}.png")

        card\_images[card\_name] = pygame.transform.scale(card\_images[card\_name], (80, 120))

large\_font = pygame.font.Font(None, 74)

medium\_font = pygame.font.Font(None, 50)

small\_font = pygame.font.Font(None, 36)

player\_cards = []

dealer\_cards = []

game\_over = False

player\_score = 0

dealer\_score = 0

player\_busted = False

dealer\_busted = False

winner = None

def deal\_card():

    suit = random.choice(suits)

    value = random.choice(values)

    return f"{value}\_of\_{suit}"

def calculate\_score(cards):

    score = 0

    ace\_count = 0

    for card in cards:

        value = card.split("\_")[0]

        if value in ['jack', 'queen', 'king']:

            score += 10

        elif value == 'ace':

            score += 11

            ace\_count += 1

        else:

            score += int(value)

    while score > 21 and ace\_count > 0:

        score -= 10

        ace\_count -= 1

    return score

def draw\_cards(cards, x, y, hide\_first=False):

    for i, card in enumerate(cards):

        if hide\_first and i == 0:

            screen.blit(card\_back, (x + i \* 100, y))

        else:

            screen.blit(card\_images[card], (x + i \* 100, y))

def display\_text(text, font, color, x, y):

    text\_surface = font.render(text, True, color)

    screen.blit(text\_surface, (x, y))

def game\_instructions():

    screen.fill(GREEN)

    display\_text("Welcome to Blackjack!", large\_font, WHITE, 100, 100)

    display\_text("Instructions:", medium\_font, WHITE, 100, 200)

    display\_text("1. Try to get as close to 21 as possible.", small\_font, WHITE, 100, 250)

    display\_text("2. Dealer must stand on 17 or higher.", small\_font, WHITE, 100, 300)

    display\_text("3. Ace can be 1 or 11.", small\_font, WHITE, 100, 350)

    display\_text("4. Press 'H' to Hit and 'S' to Stand.", small\_font, WHITE, 100, 400)

    display\_text("Click 'PLAY NOW' to start!", medium\_font, WHITE, 100, 450)

    pygame.draw.rect(screen, BLACK, (300, 500, 200, 50))

    display\_text("PLAY NOW", small\_font, WHITE, 330, 510)

    pygame.display.update()

def game\_over\_screen(winner):

    screen.fill(BLACK)

    display\_text(f"Your Score: {player\_score}", medium\_font, WHITE, 50, 50)

    draw\_cards(player\_cards, 50, 100)

    display\_text(f"Dealer's Score: {dealer\_score}", medium\_font, WHITE, 600, 50)

    draw\_cards(dealer\_cards, 600, 100)

    if winner == "Tie":

        display\_text("It's a Tie!", large\_font, WHITE, 350, 250)

    else:

        display\_text(f"{winner} Wins!", large\_font, WHITE, 350, 250)

    pygame.draw.rect(screen, GREEN, (375, 500, 200, 50))

    display\_text("PLAY AGAIN", small\_font, WHITE, 402.5, 510)

    pygame.display.update()

def reset\_game():

    global player\_cards, dealer\_cards, game\_over, player\_score, dealer\_score, player\_busted, dealer\_busted, winner

    player\_cards = []

    dealer\_cards = []

    game\_over = False

    player\_score = 0

    dealer\_score = 0

    player\_busted = False

    dealer\_busted = False

    winner = None

def main():

    global player\_cards, dealer\_cards, game\_over, player\_score, dealer\_score, player\_busted, dealer\_busted, winner

    clock = pygame.time.Clock()

    running = True

    on\_instructions = True

    while running:

        if on\_instructions:

            game\_instructions()

            for event in pygame.event.get():

                if event.type == pygame.QUIT:

                    pygame.quit()

                    sys.exit()

                if event.type == pygame.MOUSEBUTTONDOWN:

                    x, y = event.pos

                    if 300 <= x <= 500 and 500 <= y <= 550:

                        on\_instructions = False

                        reset\_game()

        else:

            screen.blit(background, (0, 0))

            if not player\_cards and not dealer\_cards:

                for \_ in range(2):

                    player\_cards.append(deal\_card())

                    dealer\_cards.append(deal\_card())

                player\_score = calculate\_score(player\_cards)

                dealer\_score = calculate\_score(dealer\_cards)

            draw\_cards(player\_cards, 400, 400)

            draw\_cards(dealer\_cards, 400, 100, hide\_first=True)

            display\_text(f"Your Score: {player\_score}", medium\_font, WHITE, 375, 350)

            display\_text(f"Dealer's Cards", medium\_font, WHITE, 375, 50)

            if game\_over:

                if winner:

                    game\_over\_screen(winner)

            pygame.display.update()

            for event in pygame.event.get():

                if event.type == pygame.QUIT:

                    pygame.quit()

                    sys.exit()

                if event.type == pygame.KEYDOWN:

                    if event.key == pygame.K\_h and not game\_over:

                        player\_cards.append(deal\_card())

                        card\_sound.play()

                        player\_score = calculate\_score(player\_cards)

                        if player\_score > 21:

                            player\_busted = True

                            winner = "Dealer"

                            lose\_sound.play()

                            game\_over = True

                    if event.key == pygame.K\_s and not game\_over:

                        while dealer\_score < 17:

                            dealer\_cards.append(deal\_card())

                            card\_sound.play()

                            dealer\_score = calculate\_score(dealer\_cards)

                        if dealer\_score > 21:

                            dealer\_busted = True

                            winner = "Player"

                            win\_sound.play()

                        elif player\_score == dealer\_score:

                            winner = "Tie"

                        elif not winner:

                            winner = "Player" if player\_score > dealer\_score else "Dealer"

                            if winner == "Player":

                                win\_sound.play()

                            else:

                                lose\_sound.play()

                        game\_over = True

                if event.type == pygame.MOUSEBUTTONDOWN and game\_over:

                    x, y = event.pos

                    if 300 <= x <= 500 and 510 <= y <= 560:

                        reset\_game()

            if player\_busted:

                winner = "Dealer"

                game\_over = True

            elif dealer\_busted:

                winner = "Player"

                game\_over = True

    clock.tick(30)

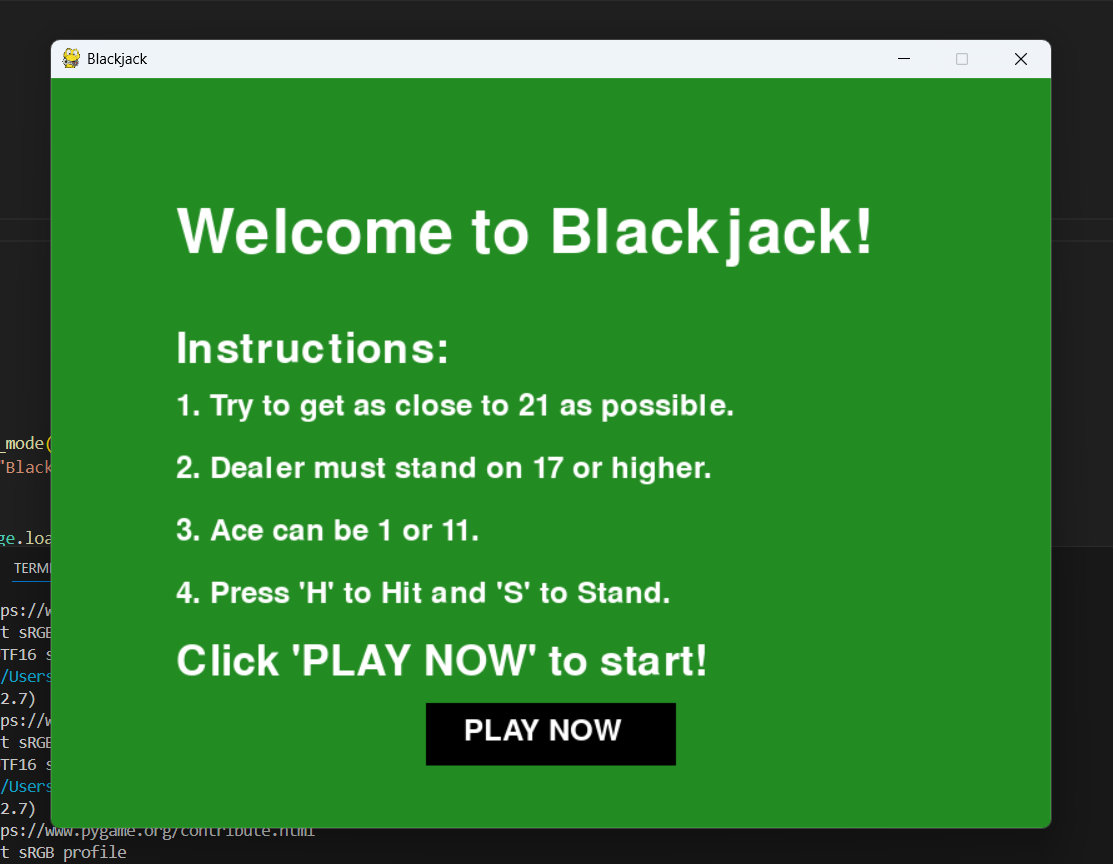
if \_\_name\_\_ == "\_\_main\_\_":

    main()

# **CHƯƠNG 5:THỬ NGHIỆM**

## 5.1. Kiểm thử chức năng chính

### 5.1.1. Giao diện hướng dẫn



Hình 5.1. Giao diện hướng dẫn

### 5.1.2. Rút bài – phím H



Hình 5.2. Rút bài – phím H

### 5.1.3. Dừng bài – phím S



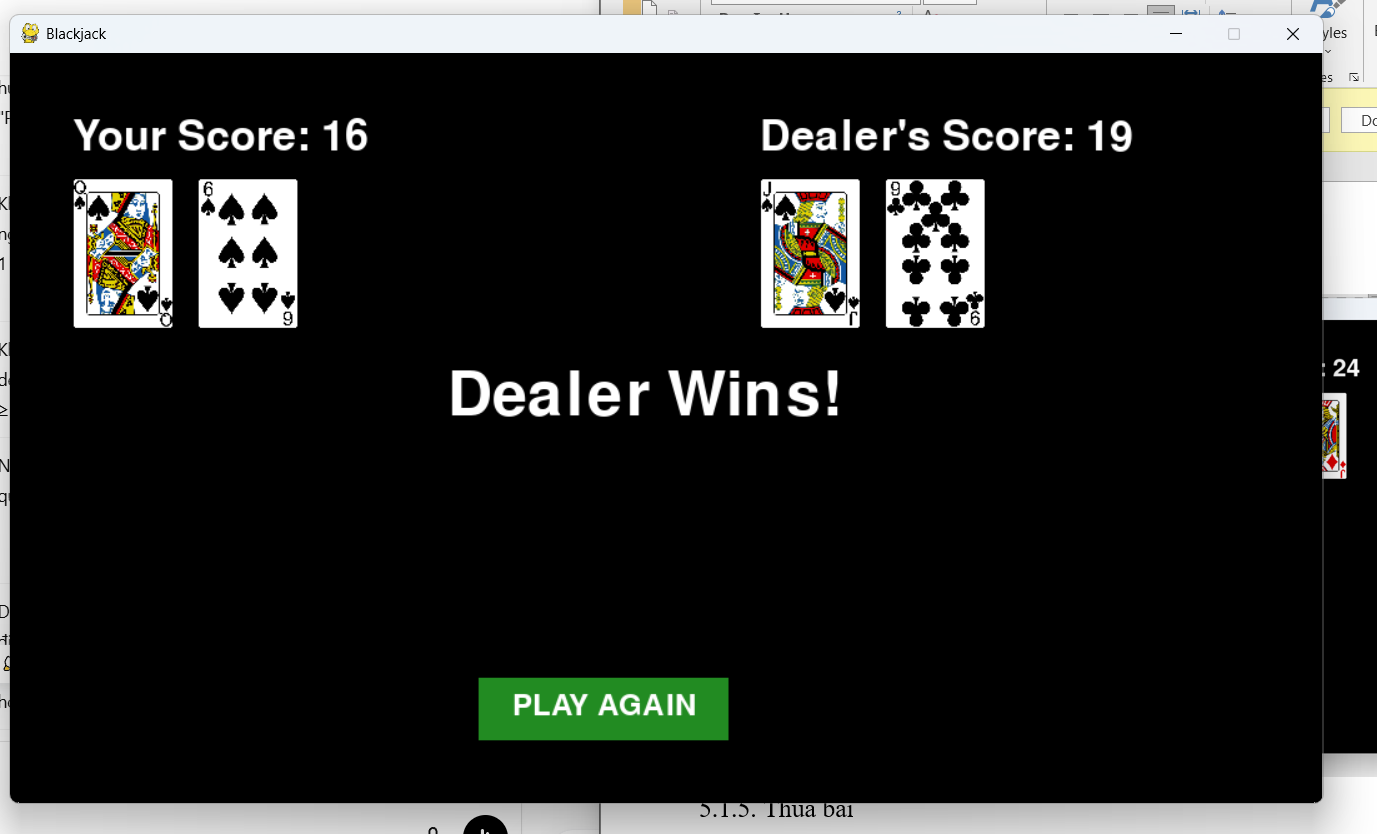
Hình 5.3. Dừng bài – phím S

### 5.1.4. Thắng bài



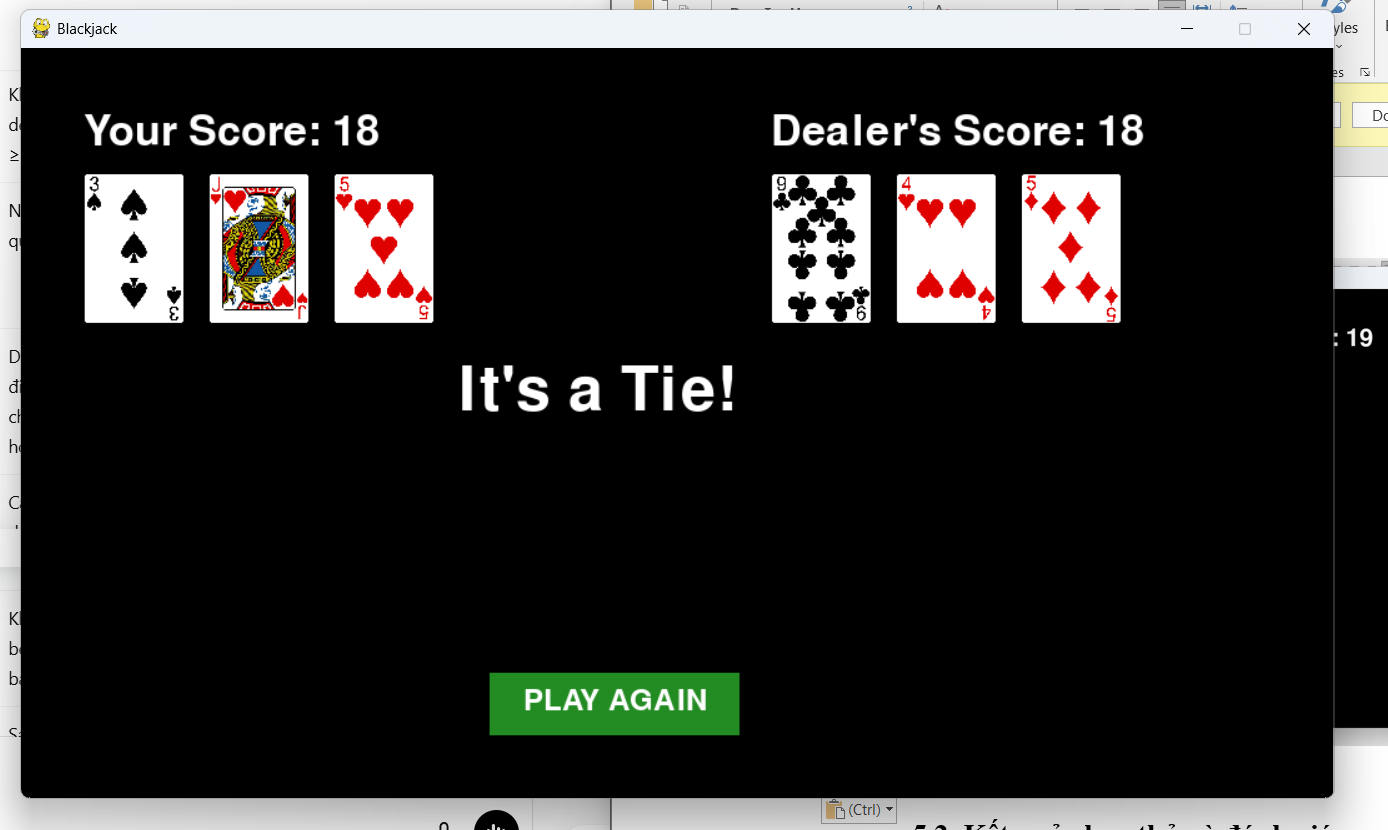
Hình 5.4. Thắng bài

### 5.1.5. Thua bài



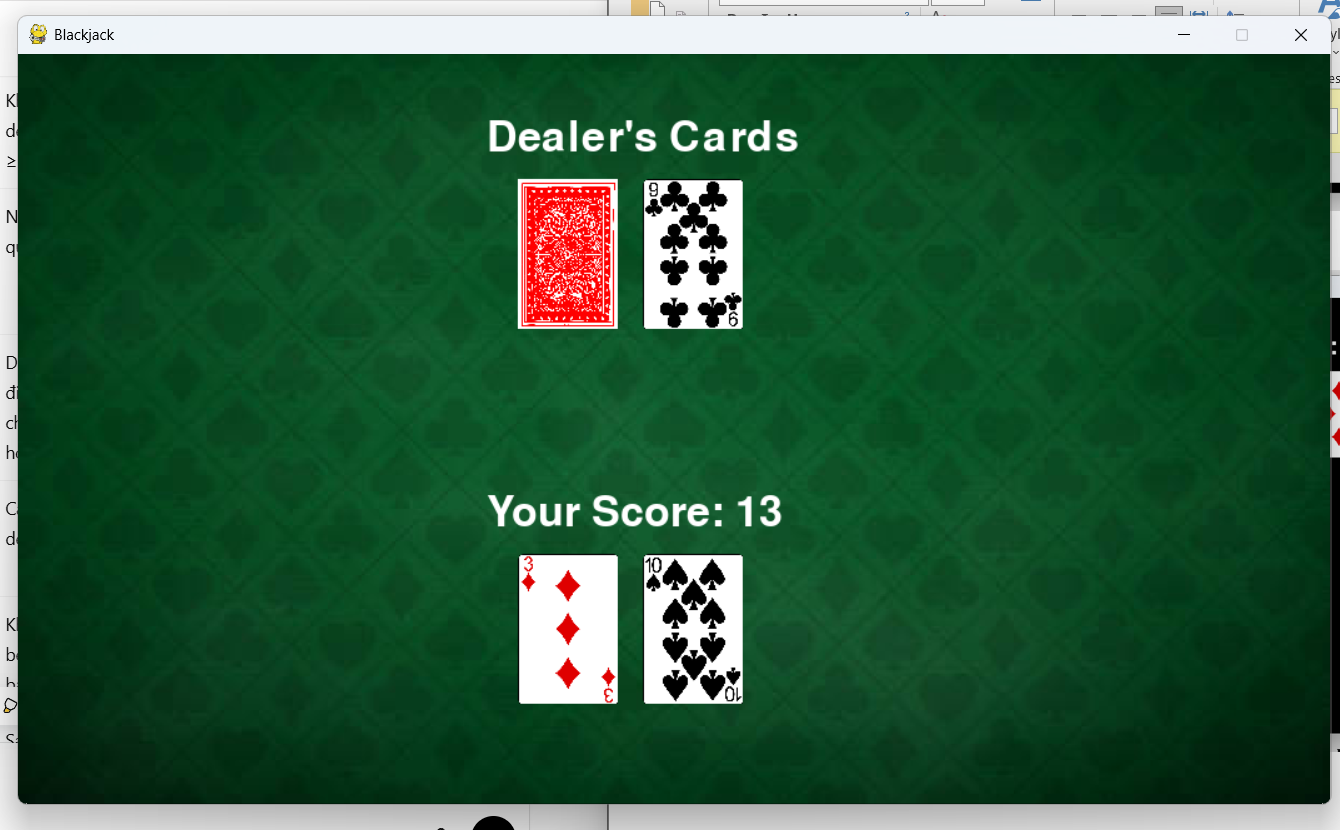
Hình 5.5. Thua bài

### 5.1.6. Hòa



Hình 5.6. Hòa

### 5.1.7. Phát bài ban đầu



Hình 5.7. Phát bài ban đầu

## 5.2. Kết quả chạy thử và đánh giá

Sau khi hoàn thiện quá trình cài đặt và kiểm thử, trò chơi Blackjack đã được chạy thử trong nhiều tình huống khác nhau để đánh giá tính ổn định và khả năng phản hồi. Kết quả như sau:

* **Khởi động game thành công**: Giao diện nền được tải lên đầy đủ cùng nhạc nền phát ngay khi bắt đầu.
* **Tải hình ảnh và âm thanh không lỗi**: Các lá bài được hiển thị rõ ràng, hình nền và hiệu ứng âm thanh hoạt động bình thường.
* **Giao diện hướng dẫn rõ ràng**: Người chơi dễ dàng nắm được luật chơi qua phần giới thiệu ban đầu.
* **Chức năng rút bài (Hit) và dừng bài (Stand) vận hành chính xác**: Điểm số được cập nhật đúng với từng lá bài rút ra, xử lý tình huống vượt quá 21 điểm nhanh và chính xác.
* **Xử lý thắng – thua – hòa hợp lý**: Trò chơi đưa ra kết quả đúng với luật chơi và thể hiện kết quả rõ ràng trên màn hình.
* **Chức năng chơi lại sau khi kết thúc game hoạt động tốt**: Game reset về trạng thái ban đầu mà không bị lỗi.
* **Không có hiện tượng treo, đơ hay thoát bất ngờ trong quá trình chơi thử nghiệm**.

### **Đánh giá tổng thể**

* **Tính ổn định**: Game hoạt động ổn định trong suốt quá trình thử nghiệm trên nhiều thiết bị có cấu hình trung bình.
* **Tính trực quan**: Giao diện đơn giản, dễ sử dụng, hình ảnh rõ ràng, màu sắc hài hòa phù hợp với chủ đề casino.
* **Trải nghiệm người dùng**: Game mang lại trải nghiệm thú vị nhờ âm nhạc nền sống động, hiệu ứng rút bài chân thực và phản hồi nhanh.
* **Tính năng**: Đáp ứng đầy đủ các yêu cầu cơ bản của trò chơi Blackjack (chia bài, tính điểm, xử lý thắng thua, âm thanh, hình ảnh...).

# **CHƯƠNG 6:KẾT LUẬN**

## ****6.1. Tổng kết nội dung báo cáo****

Trong quá trình thực hiện đề tài, em đã tiến hành nghiên cứu và xây dựng thành công trò chơi Blackjack sử dụng ngôn ngữ lập trình Python cùng thư viện Pygame. Báo cáo đã trình bày đầy đủ các nội dung từ phân tích yêu cầu, thiết kế hệ thống, triển khai code, đến kiểm thử và đánh giá. Trò chơi đáp ứng được các chức năng cơ bản như chia bài, tính điểm, xử lý các thao tác Hit/Stand, xác định thắng thua, và có giao diện đồ họa thân thiện với người dùng.

## ****6.2. Đánh giá kết quả đạt được****

* Hệ thống hoạt động ổn định, giao diện trực quan.
* Các luật chơi cơ bản của Blackjack được áp dụng đúng.
* Có tích hợp hiệu ứng hình ảnh, âm thanh giúp tăng trải nghiệm người chơi.
* Mã nguồn được tổ chức rõ ràng theo module giúp dễ bảo trì và mở rộng.
* Tuy nhiên, vẫn còn một số hạn chế:
* Trò chơi chỉ hỗ trợ 1 người chơi đấu với máy (Dealer).
* Chưa có chế độ chơi nhiều người hoặc lưu lại lịch sử chơi.

## ****6.3. Hướng phát triển trong tương lai****

Trong tương lai, trò chơi có thể được nâng cấp và mở rộng thêm các tính năng sau:

* Thêm chế độ nhiều người chơi (Multiplayer).
* Tối ưu hiệu suất và hỗ trợ chơi trên các thiết bị di động.
* Thêm bảng xếp hạng và lưu lịch sử chơi.
* Cải thiện AI của Dealer để trò chơi trở nên hấp dẫn hơn.

## ****6.4. Kết luận chung****

Đề tài đã giúp emcố và vận dụng kiến thức về lập trình hướng đối tượng, thiết kế giao diện, xử lý sự kiện và kiểm thử phần mềm. Việc hoàn thiện trò chơi Blackjack không chỉ giúp nâng cao kỹ năng lập trình mà còn tạo ra một sản phẩm thực tế có tính giải trí cao. Đây là nền tảng để phát triển các dự án game hoặc ứng dụng lớn hơn trong tương lai.

# **TÀI LIỆU THAM KHẢO**

[1] Guru99. (n.d.). *Manual Testing Checklist*.  
https://www.guru99.com/manual-testing-checklist.html

[2] Real Python. (n.d.). *Debugging and Testing Python Code*.  
https://realpython.com/python-debugging-pdb

[3] GameDev.net. (n.d.). *Game Testing Best Practices*.  
<https://short.com.vn/yvmy>

[4]AskPython.(n.d.).BlackjackGameinPython(SourceCode).  
https://www.askpython.com/python/examples/blackjack-game

[5]CodeReviewStackExchange.(n.d.).BlackjackinPygame.  
<https://codereview.stackexchange.com/questions/281356/blackjack-using-pygame>

[6]PygameDocs.(n.d.).PygameDocumentation.  
https://www.pygame.org/docs/