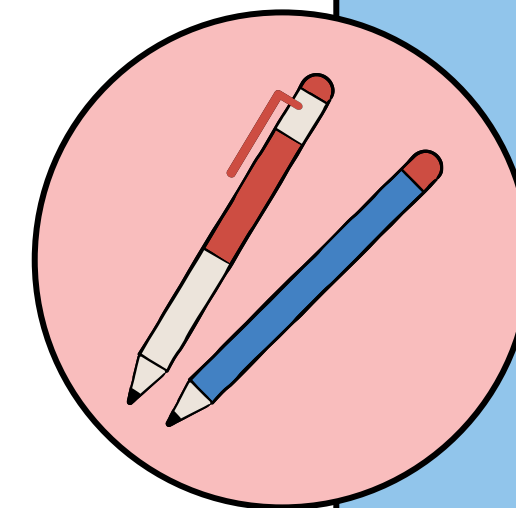
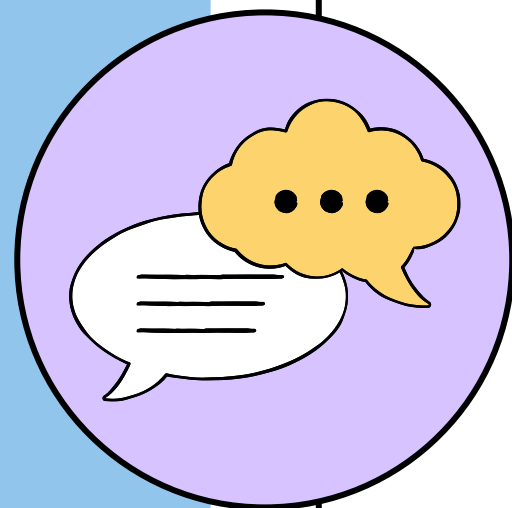


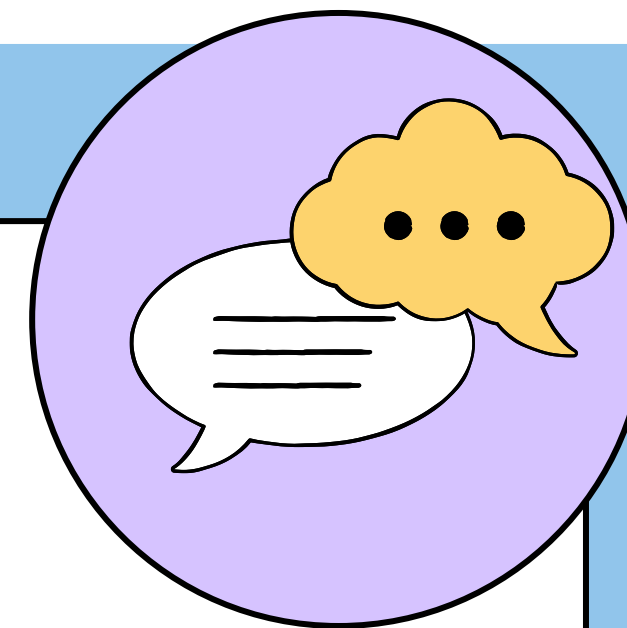
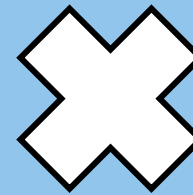
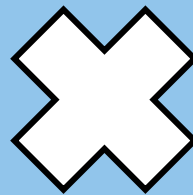


## BÀI THUYẾT TRÌNH

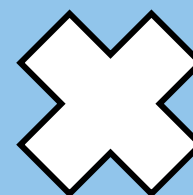
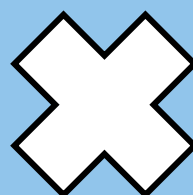


# Môn học: **Lập trình Python**

Sinh viên : Lương Văn Học  
MSSV: K225480106025



**TÊN ĐỀ TÀI: XÂY GAME BLACKJACK  
VỚI GUI**



# Giới thiệu đề tài

## ❖ Mục tiêu:

Xây dựng trò chơi Blackjack với giao diện người dùng (GUI) sử dụng ngôn ngữ lập trình Python.

## ❖ Yêu cầu chính:

- Sử dụng các lớp: BJ\_Card, BJ\_Deck, BJ\_Hand, BJ\_Game.
- Giao diện hiển thị: Lá bài, điểm số, kết quả (thắng/thua/hòa).
- Xử lý các hành động: Hit, Stand, kiểm tra Blackjack, Bust.

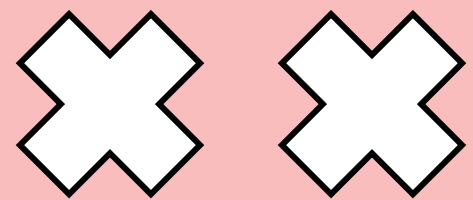
## ❖ Công cụ sử dụng:

**Python:** Ngôn ngữ lập trình chính.

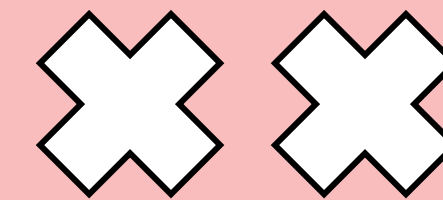
**tkinter:** Thư viện xây dựng giao diện người dùng.

**pygame:** Tích hợp âm thanh (hiệu ứng Hit, Win, Lose, Tie).

**Pillow:** Xử lý và hiển thị hình ảnh lá bài.



# Tổng quan hệ thống

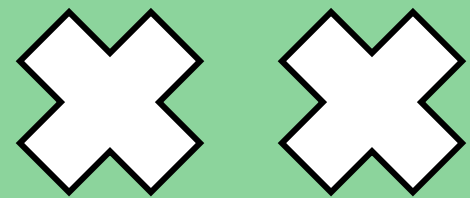


## ❖ Cấu trúc hệ thống:

- Các mô-đun chính: BJ\_Card, BJ\_Deck, BJ\_Hand, BJ\_Game, GUI.
- Tương tác: Người chơi sử dụng nút “Hit” và “Stand” để điều khiển trò chơi.
- Kết quả: Hiển thị lá bài, điểm số, thông báo thắng/thua/hòa.

## ❖ Quy trình hoạt động:

- Chia bài: 2 lá cho người chơi và nhà cái.
- Người chơi chọn “Hit” (rút bài) hoặc “Stand” (dừng).
- Nhà cái rút bài nếu điểm  $< 17$ .
- So sánh điểm, hiển thị kết quả ván bài.



## Mô-đun chính

### ❑ BJ\_Card:

- Mô phỏng lá bài với thuộc tính rank (2-A) và suit (Hearts, Spades, Diamonds, Clubs).
- Phương thức `get_value()`: Tính điểm (J/Q/K=10, A=1 hoặc 11).

### ❑ BJ\_Deck:

- Quản lý bộ bài 52 lá.
- Chức năng: Xáo bài ngẫu nhiên, rút bài bằng phương thức `deal()`.

### ❑ BJ\_Hand:

- Quản lý tay bài của người chơi và nhà cái.
- Phương thức: `add_card()` (thêm bài), `get_value()` (tính điểm, điều chỉnh A).

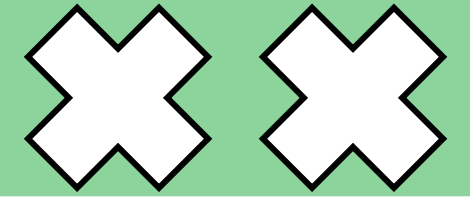
### ❑ BJ\_Game:

- Điều khiển logic: Chia bài, xử lý Hit/Stand, xác định kết quả.

### ❑ GUI:

- Hiển thị lá bài, điểm số, kết quả.
- Nút bấm: “Hit”, “Stand”, “New Game”.

# Quy trình phát triển



**Bước 1:** Thiết kế các lớp (BJ\_Card, BJ\_Deck, BJ\_Hand, BJ\_Game) để mô phỏng lá bài, bộ bài, tay bài, và logic trò chơi.

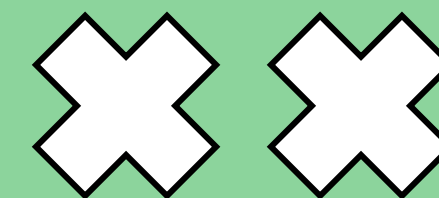
**Bước 2:** Xây dựng giao diện người dùng (GUI) với thư viện tkinter (frame, nút bấm, label).

**Bước 3:** Tích hợp âm thanh (pygame) và hình ảnh (Pillow) cho lá bài, hiệu ứng.

**Bước 4:** Xử lý logic: Chia bài, tính điểm, kiểm tra Blackjack/Bust.

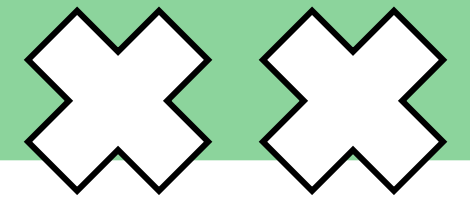
**Bước 5:** Kiểm thử tính năng (chia bài, rút bài, kết quả) và tối ưu hóa hiệu suất.

# Giao diện người dùng (GUI)



- ❖ **Công cụ:** Thư viện tkinter của Python.
- ❖ **Thành phần chính:**
  - **Frame:** Hiển thị lá bài của người chơi và nhà cái.
  - **Nút bấm:** “Hit” (rút bài), “Stand” (dừng), “New Game” (bắt đầu ván mới).
  - **Label:** Hiển thị điểm số hiện tại và kết quả ván bài (thắng/thua/hòa).
- ❖ **Tính năng:**
  - Cập nhật giao diện theo hành động: Rút bài, kết thúc ván.
  - Hiển thị lá bài dưới dạng hình ảnh/icon.

# Tích hợp đa phương tiện



- ❖ **Âm thanh (pygame):**

Hiệu ứng: Hit, Win, Lose, Tie.

Phát khi người chơi rút bài hoặc khi ván bài kết thúc.

- ❖ **Hình ảnh (Pillow):**

Hiển thị lá bài dưới dạng icon/image.

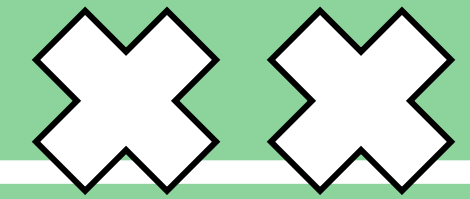
Lá bài úp của nhà cái được hiển thị khi người chơi chưa nhấn “Stand”.

- ❖ **Mục tiêu:** Tăng tính sinh động và trải nghiệm thực tế cho trò chơi.

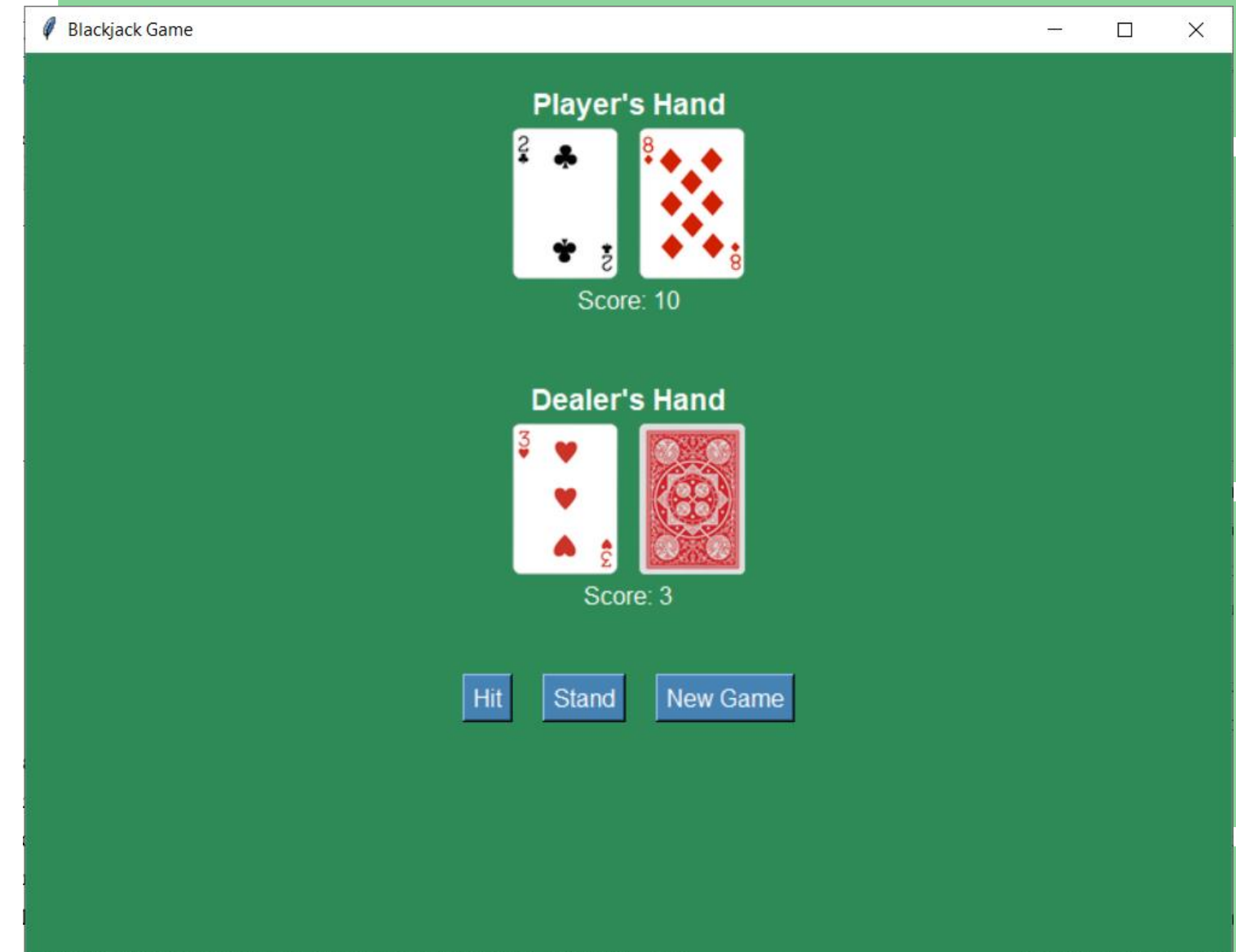
- ❖ **So sánh kết quả:** Sau khi người chơi chọn "Stand", nhà cái sẽ mở bài và rút thêm nếu cần. Kết quả trò chơi được xác định thông qua việc so sánh điểm giữa người chơi và nhà cái.



# Kết quả thực nghiệm - Màn hình ban đầu

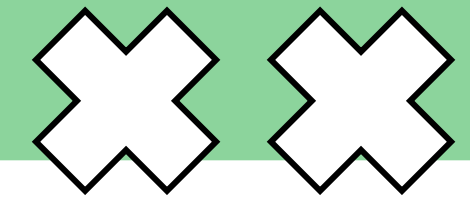


- ❖ **Mô tả:** Giao diện khi bắt đầu một ván bài mới.
- ❖ **Hiển thị:**
  - Người chơi: 2 lá bài lật ngửa.
  - Nhà cái: 1 lá bài lật ngửa + 1 lá bài úp.
- ❖ **Chức năng:**
  - Lá bài hiển thị dưới dạng hình ảnh/icon.
  - Điểm số ban đầu được cập nhật trên giao diện.



*Màn hình khi bắt đầu ván bài*

# Kết quả thực nghiệm - Hành động “Hit”

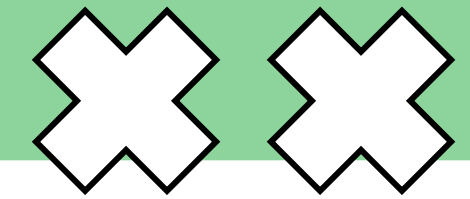


- ❖ **Mô tả:** Người chơi nhấn nút “Hit” để rút thêm bài.
- ❖ **Kết quả:**
  - Lá bài mới được thêm vào tay bài người chơi.
  - Điểm số được cập nhật ngay trên giao diện.
  - Kiểm tra tình trạng Bust ( $>21$ ): Nếu xảy ra, hiển thị thông báo thua.



*Màn hình khi người chơi nhấn "Hit"*

# Kết quả thực nghiệm - Hành động “Stand” và Kết quả



❖ **Mô tả:** Người chơi nhấn “Stand”, chuyển lượt sang nhà cái.

❖ **Kết quả:**

Nhà cái mở bài, rút thêm nếu điểm  $< 17$ .

So sánh điểm:

Người chơi thắng nếu điểm cao hơn và  $\leq 21$ .

Nhà cái thắng nếu điểm cao hơn và  $\leq 21$ .

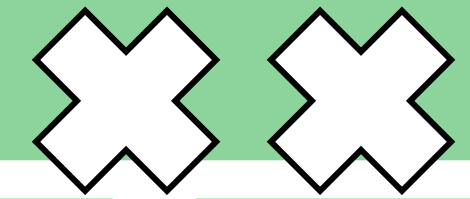
Hòa nếu điểm bằng nhau.

Hiển thị thông báo: “Win”, “Lose”, hoặc “Tie”.



*Màn hình khi người chơi nhấn "Stand"*

# Đánh giá



## ❖ Thành công:

- Tính năng: Chia bài, tính điểm, xử lý Blackjack/Bust hoạt động chính xác.
- Giao diện: Đơn giản, dễ sử dụng, phản hồi nhanh với hành động người chơi.
- Tích hợp: Âm thanh và hình ảnh hiển thị mượt mà.

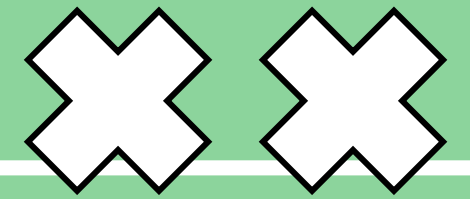
## ❖ Hạn chế:

- Tốc độ xử lý cần tối ưu khi rút nhiều bài liên tiếp.
- Hiệu ứng đồ họa và âm thanh còn đơn giản, chưa phong phú.



*Màn hình hiển thị kết quả trò chơi*

# Kết luận và cải tiến

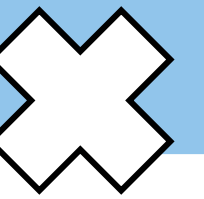
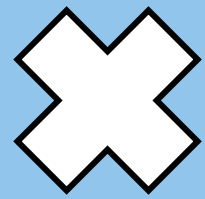


## ❖ Kết luận:

- Hoàn thành trò chơi Blackjack với đầy đủ tính năng: chia bài, tính điểm, xử lý Blackjack/Bust.
- Ứng dụng thành công: OOP, tkinter (GUI), pygame (âm thanh), Pillow (hình ảnh).
- Rèn luyện kỹ năng: Lập trình game, quản lý trạng thái, tích hợp đa phương tiện.

## ❖ Cải tiến đề xuất:

- Nâng cấp hiệu ứng đồ họa và âm thanh cho trải nghiệm sinh động hơn.
- Tối ưu tốc độ xử lý khi rút nhiều bài.
- Thêm chế độ đa người chơi và tính năng lưu điểm/thống kê ván bài.



THANK YOU

