

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM**

---□&□---



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN THỊ GIÁC MÁY TÍNH

Đề tài:

Powerpoint Animation

***Giáo Viên Hướng Dẫn:
Thầy Nguyễn Vinh Tiệp***

Sinh viên thực hiện:		
STT	Họ tên	MSSV
1	Nguyễn Lộc Linh	19521754
2	Trần Duy Quang	19522102
3	Lê Dương Khánh Việt	19522515

Thành phố HCM, 06/2021

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

LỜI CẢM ƠN

Sau quá trình học tập và rèn luyện tại trường Đại học Công nghệ Thông tin chúng em đã được trang bị các kiến thức cơ bản, các kỹ năng thực tế để có thể hoàn thành đồ án môn học của mình.

Chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Vinh Tiệp đã quan tâm, hướng dẫn, truyền đạt những kiến thức và kinh nghiệm cho chúng em trong suốt thời gian học tập môn Nhập môn thị giác máy tính.

Trong quá trình làm đồ án môn không tránh khỏi được những sai sót, chúng em mong nhận được sự góp ý của quý cô và các bạn để được hoàn thiện hơn.

TP. Hồ Chí Minh, tháng 6, năm 2021.

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN	5
1.1. Giới thiệu	5
1.2. Mục đích và yêu cầu của đề tài	5
1.2.1. Mục đích.....	5
1.2.2. Yêu cầu.....	5
CHƯƠNG 2: NỘI DUNG KIẾN THỨC	6
2.1. Các loại hệ màu thường gặp	6
2.1.1. Hệ màu RGB	6
2.1.2. CMYK.....	7
2.1.3. HSV.....	7
2.1.4. YCbCr (Y'CbCr).....	8
2.2. Affine Transformation	9
2.3. Transparent Image	11
2.4. Các animation đã làm	12
2.5. Phương pháp thực hiện chung	19
2.6. Source code python và demo	20
Chương 3: CÀI ĐẶT VÀ KIỂM THỬ	21
3.1 Danh sách các animation	21
3.2 Mức độ hoàn thành các animation	21
Chương 4: NHẬN XÉT VÀ KẾT LUẬN	22
4.1 Nhận xét	22
4.1.1 Kết quả đạt được.....	22
4.1.2 Hạn chế.....	22
4.1.3 Kết luận.....	22
4.2 Hướng phát triển	22
Chương 5: TÀI LIỆU THAM KHẢO	23
Chương 6: BẢNG PHÂN CÔNG CÔNG VIỆC	24

CHƯƠNG 1: TỔNG QUAN

1.1. Giới thiệu

PowerPoint là một phần mềm trình chiếu trong bộ công cụ Microsoft Office, được đông đảo học sinh, giáo viên hay giới văn phòng sử dụng cho những bài thuyết trình của mình. Cùng với Microsoft Word và Microsoft Excel, Microsoft PowerPoint đã trở thành một thiết bị văn phòng không thể thiếu. Nó cho phép người dùng tạo ra những Slide để thể hiện những thông điệp với các hiệu ứng đẹp. Và để tạo ra các hiệu ứng đó thì animation là một thứ không thể thiếu.

1.2. Mục đích và yêu cầu của đề tài

1.2.1. Mục đích

- Tạo ra các animation mới và sẵn có của powerpoint bằng việc code python
- Kết quả animation tạo ra có chức năng và hiệu ứng giống hoặc gần giống so với animation của powerpoint
- Nghiên cứu và cải tiến animation, các hiệu ứng animation mới cho powerpoint

1.2.2. Yêu cầu

- Giao diện và bố cục sắp xếp hợp lí, rõ ràng cho người dùng sử dụng
- Bộ phận quản lí có thể điều chỉnh, bảo trì code.
- Animation làm ra phải tương đối đẹp so với animation có sẵn của powerpoint

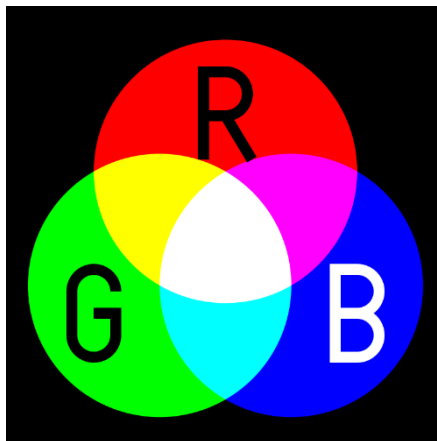
CHƯƠNG 2: NỘI DUNG KIẾN THỨC

2.1. Các loại hệ màu thường gặp

Powerpoint animaiton cũng có những hoạt ảnh thay đổi màu sắc. Mà màu sắc thì cũng có nhiều mô hình màu khác nhau. Lựa chọn hệ màu phù hợp có thể giúp chương trình dễ thực hiện hơn, ít bị lệch màu, ... Vì vậy ta sẽ đi qua 1 vài hệ màu để có thể lựa chọn sử dụng trong chương trình.

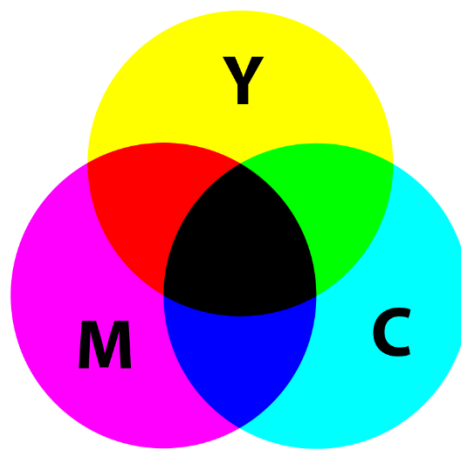
2.1.1. Hệ màu RGB

Màu RGB là chế độ hiển thị màu tự nhiên ở hầu hết các màn hình điện tử thời nay. RGB là viết tắt của 3 màu cơ bản Red, Green, Blue. Trong đó 3 màu này được kết hợp với nhau theo nhiều cách để tạo ra các màu khác. Hệ màu RGB phụ thuộc vào các thiết bị điện tử để có hiển thị nên nhiều thiết bị điện tử khác nhau có thể cho ra màu khác nhau dù là cùng thể hiện chung 1 giá trị RGB, thậm chí là cùng 1 thiết bị điện tử nhưng khi cũ đi cũng cho ra màu khác nhau. Hệ màu này được sử dụng tốt nhất cho việc thiết kế web, hình ảnh kỹ thuật số, ...



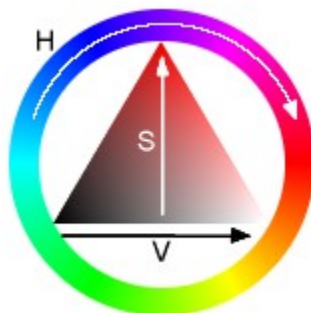
2.1.2. CMYK

CMYK (đôi khi là YMCK) là mô hình màu loại trừ sử dụng trong in ấn, khác với hệ màu RGB là mô hình màu bổ sung. Tức là màu mình thấy là màu được phản xạ lại và những màu khác bị hấp thụ. Như vậy trong CMYK, khi trộn 3 màu xanh lơ (Cyan), màu hồng sẫm (Magenta) và màu vàng (Yellow) lại với nhau thì sẽ ra màu đen. Nhưng màu đen này không thực sự là màu đen nên người ta cho thêm mực đen vào hệ màu này. Đó là viết tắt của Key tức màu đen.



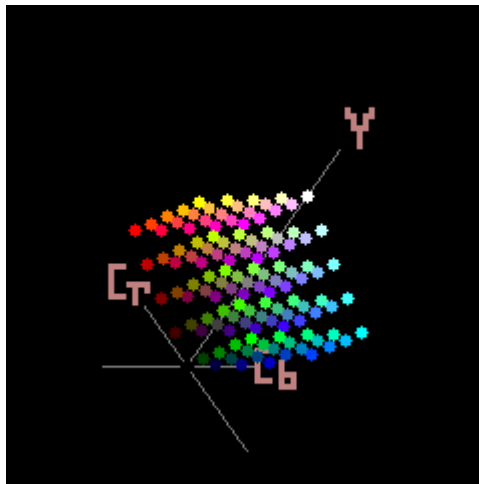
2.1.3. HSV

HSV (hay HSB) là viết tắt của Hue, Saturation và Value (Brightness). Đây là hệ màu có thể sử dụng để thể hiện màu sắc bên cạnh hệ màu RGB. Với Hue là vùng màu ở bên ngoài; Saturation là độ bão hòa màu (S càng lớn, màu càng rõ); Value là độ sáng, khi value về 0 thì thành màu đen.



2.1.4. YCbCr (Y'CbCr)

YCbCr là một họ không gian màu được sử dụng trong các đường dẫn video và ảnh kỹ thuật số. Trong đó, Y là thành phần thay đổi độ sáng của ảnh; Cb là độ khác nhau của màu xanh dương; Cr là độ khác nhau của màu đỏ. Các không gian màu Y'CbCr được xác định bằng phép biến đổi tọa độ toán học từ không gian màu RGB tương ứng. Do đó Y'CbCr có các tính chất của RGB. Y'CbCr thích hợp hơn RGB ở lưu trữ và truyền tải dữ liệu, do mắt người nhạy cảm hơn trong việc phân biệt màu đen trắng hơn là màu sắc khác. Vì thế Y'CbCr truyền tải dữ liệu bằng cách tách Y riêng và CbCr riêng. Y sẽ được truyền đầy đủ thông tin trong khi thông tin của CbCr có thể được giảm, nén, ghép hay được xử lý riêng để cải thiện hiệu quả hệ thống.



Kết luận:

- Sau khi đã đi qua 1 vài hệ màu thì ta thấy hệ **RGB** và **HSV** là lựa chọn hợp lý nhất, bởi lẽ hệ màu **CYMK** dùng trong việc in ấn thì không phù hợp cho việc đổi màu hình ảnh, thậm chí có thể gây sai lệch màu sắc. Vì hầu hết animation được biểu diễn trên màn hình máy tính nên cần thiết phải dùng đến hệ màu **RGB** để đọc ảnh. Tại sao lại có cần thêm hệ màu **HSV** trong khi đã có **RGB** rồi? Bởi vì hệ màu **HSV** có 1 đặc điểm mà **RGB** khó có thể làm được đó là biến đổi màu và chọn vùng màu. Giả sử với hiệu ứng swipe, ta thấy ảnh sẽ trắng dần. Với **RGB**, việc thay đổi các giá trị từ 1 màu sang màu trắng không đồng đều với các màu khác nhau. Còn đối với hệ màu **HSV**, cùng 1 việc như vậy nhưng dù là màu gì thì chỉ cần làm 2 bước để 1 màu tiến tới màu trắng đó là tăng Value và giảm Saturation. Nhờ vậy mà việc code chương trình trở nên đơn giản hơn. Còn **YCbCr** thích hợp cho việc lưu ảnh hơn và truyền tải dữ liệu hơn là dùng để xử lý ảnh.

2.2. Affine Transformation

- **Affine Transformation** là phép biến đổi tọa độ điểm đặc trưng của đối tượng thành tập tương ứng với các điểm mới để tạo ra hiệu ứng cho toàn đối tượng. Phép biến đổi Affin (Affine transformation) T ánh xạ tập P(x, y) sang tập Q(x', y'):

- với a, b, c, d, e, f là các hệ số
$$\begin{cases} x' = ax + by + c \\ y' = dx + ey + f \end{cases}$$

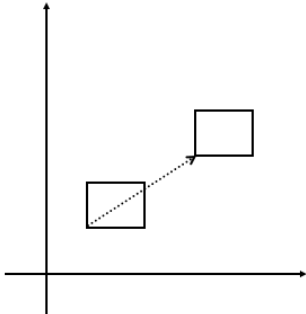
- Biểu diễn ma trận:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Leftrightarrow Q = T.P$$

- **Các phép biến đổi cơ sở:**

a) Phép tịnh tiến:

Dùng để dịch chuyển đối tượng từ vị trí này sang vị trí khác



Gọi $tr = (tr_x, tr_y)$ là vector tịnh tiến từ điểm P đến điểm Q thì:

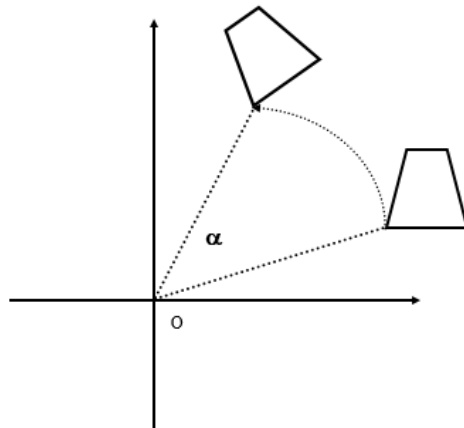
$$\begin{cases} x' = x + tr_x \\ y' = y + tr_y \end{cases}$$

Ma trận biến đổi của phép tịnh tiến.

$$T(tr_x, tr_y) = \begin{pmatrix} 1 & 0 & tr_x \\ 0 & 1 & tr_y \\ 0 & 0 & 1 \end{pmatrix}$$

b) Phép quay:

- Là phép biến đổi nhằm thay đổi hướng của đối tượng. Một phép quay (Rotation) cần phải có: Tâm quay và Góc quay (Quy ước ngược chiều kim đồng hồ)



$$\begin{cases} x' = \cos \alpha x - \sin \alpha y \\ y' = \sin \alpha x + \cos \alpha y \end{cases} \Leftrightarrow T(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

c) Phép biến đổi tỉ lệ: (Scaling)

- Là phép biến đổi giúp co giãn đối tượng

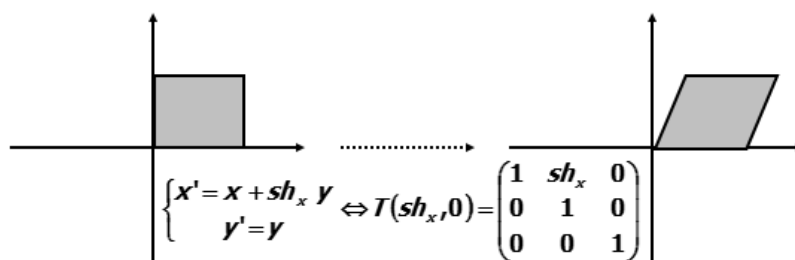
$$\begin{cases} x' = s_x x \\ y' = s_y y \end{cases} \quad T(s_x, s_y) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Với s_x và s_y được gọi là hệ số co giãn theo trục Ox và trục Oy ($s_x, s_y \leq 1$)

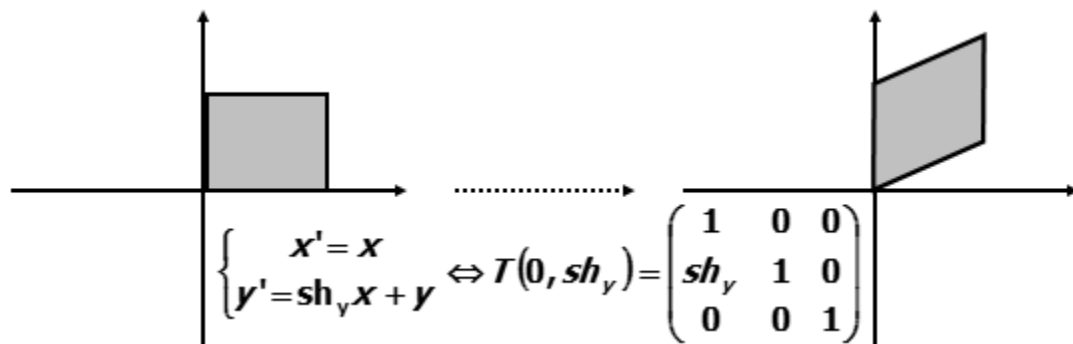
d) Phép biến dạng: (Shearing)

- Là phép biến đổi thay đổi hình dạng của đối tượng.

- Phép biến dạng theo trục x làm thay đổi hoành độ còn tung độ giữ nguyên.



- Phép biến dạng theo trục y làm thay đổi tung độ còn hoành độ giữ nguyên.



- Phép biến đổi dạng tổng quát.

$$\begin{cases} x' = x + sh_x y \\ y' = sh_y x + y \end{cases} \Leftrightarrow T(sh_x, sh_y) = \begin{pmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2.3. Transparent Image

- Trong thị giác máy tính, **Transparent Image** có thể được định nghĩa là một tấm ảnh có một phần hoặc toàn bộ đã được xóa (hay có thể nói phần đó đã hoàn toàn “trong suốt”) và có thể đưa hiệu ứng của bất kỳ tấm ảnh nào khác vào tấm hình.

- **Transparency** có thể được xem là một “phần mở rộng” của một tấm ảnh. Nó chỉ được áp dụng trong một số loại file (Vd như GIF, PNG, TIFF,...) bằng cách sử dụng một **Transparent Color** hoặc sử dụng **Alpha channel** ([Alpha Compositing](#))

- **Transparency** có thể được sử dụng trên Raster image (hay digital image) và vector image. Tuy nhiên, chúng ta chỉ nói về Transparency trên Raster image, hay cụ thể hơn là **Digital image**. Với mỗi pixel được biểu diễn trong một hệ màu, sẽ có một giá trị mà ở đó ta sẽ định nghĩa là “giá trị trong suốt” và nó không phải là một màu trong thực tế. Nếu bộ giải mã màu sắc gặp pixel mang giá trị này, nó sẽ được thể hiện bằng màu nền của phần màn hình mà tấm ảnh đặt lên. Để dễ hiểu, hãy tưởng tượng một Transparent image là một tấm ảnh với toàn bộ pixel nhận giá trị trong suốt thì tấm ảnh này trông giống tấm gương vậy, nó sẽ cho ta thấy những pixel của background mà tấm ảnh này được đặt lên

- Với một tấm ảnh, có thể có nhiều cách biểu diễn màu sắc trên các hệ màu khác nhau. Tuy nhiên, để có thể tạo ra một transparent image, ta có thể sử dụng kênh **màu rgb** kết hợp với **Alpha channel** (rgba), với giá trị alpha trong khoảng 0.0 (fully transparent) và 1.0 (not transparent at all). Đây là một trong số nhiều cách để tạo một Transparent Image. Với mỗi hệ màu khác nhau, ta có thể hoặc thêm một kênh màu alpha vào hệ màu đó hoặc set giá trị mà tại đó nếu pixel mang giá trị này thì sẽ thành transparent pixel.

- **Ví dụ:** Ta có thể tạo một image hệ màu rgba. Ở mỗi vị trí pixel, ta mong muốn nó trở thành transparent pixel, ta sẽ set giá trị alpha của rgba bằng 1. Khi đó, tấm ảnh đó sẽ trở thành Transparent Image.

* Ứng dụng của Transparent Image trong Powerpoint Animation:

- Ta có thể set một tấm ảnh với hệ màu thông thường (vd rgb) và thêm kênh màu alpha (trở thành rgba), ta có thể cho tấm ảnh đó làm background (Transparent background) và áp dụng hiệu ứng animation lên tấm ảnh đó

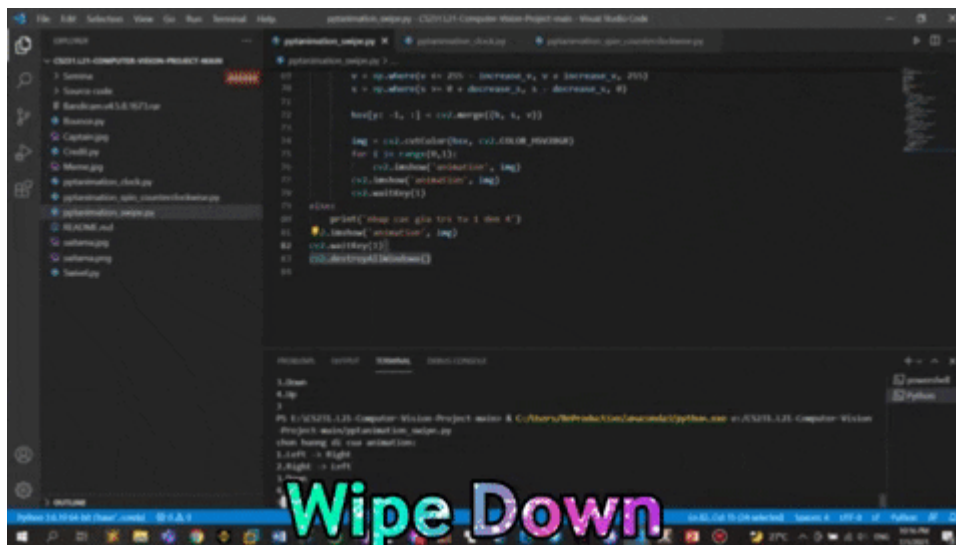
2.4. Các animation đã làm

Các animation đã làm của nhóm dù không nhiều nhưng chúng cách hoạt động của chúng gần tương tự các animation khác. Do đó có thể áp dụng phương pháp thực hiện của 1 hoạt ảnh cho 1 hoạt ảnh khác tương tự.

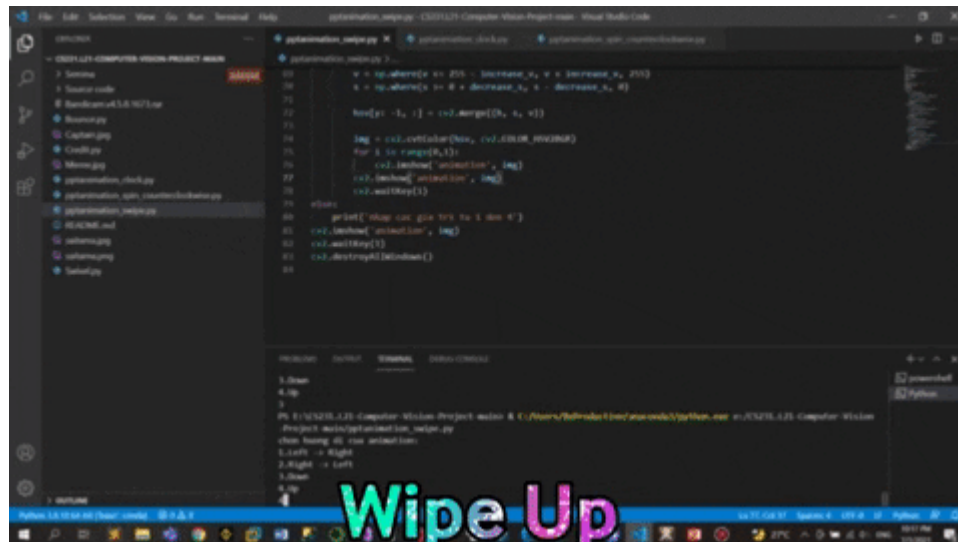
2.4.1. Wipe

Ở animation wipe down ta thấy ảnh sáng dần từ trên xuống và chuyển sang màu trắng. Phương pháp thực hiện của animation này là chuyển ảnh sang dạng HSV, duyệt ảnh theo các dòng từ trên xuống và trừ dần Saturation của các dòng cho đến khi bằng 0, đồng thời tăng Value lên dần đến 100. Mỗi lần thực hiện thay đổi như vậy thì sẽ xuất ảnh ra màn hình. Tổng hợp các ảnh như vậy tạo thành một hiệu ứng wipe down.

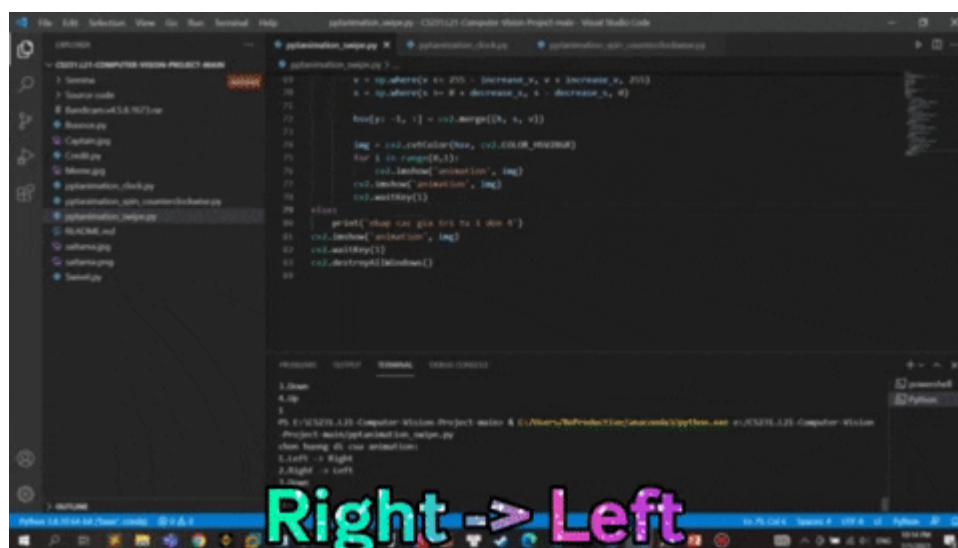
Tương tự như vậy với các animation Wipe up, Wipe left, Wipe right. Thậm chí có thể áp dụng với các animation Fade, Split, Shape.



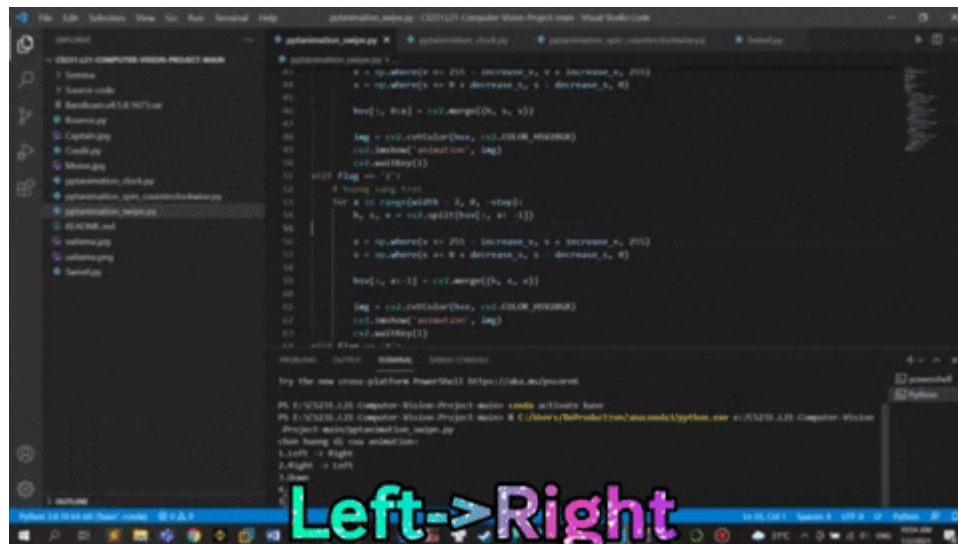
Hình 5.4. 1. Hiệu ứng Wipe down



Hình 5.4. 2. Hiệu ứng Wipe up



Hình 5.4. 3. Hiệu ứng Wipe Left

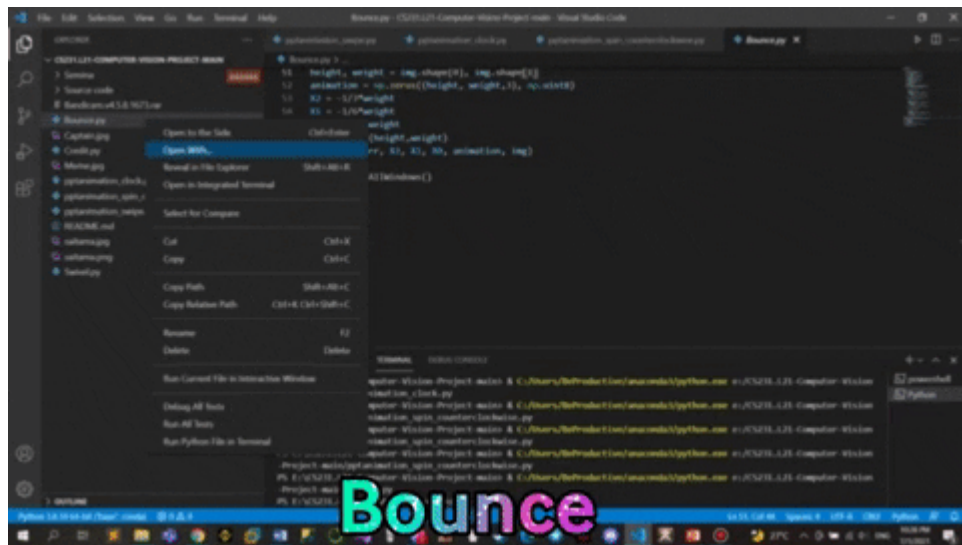


Hình 5.4. 4. Hiệu ứng Wipe right

2.4.2. Bounce

- Đối với Bounce: việc cần làm là tạo ra 1 tập ảnh ở các vị trí khác nhau. Khi đó xuất tập ảnh trên ta sẽ thu được hiệu ứng nảy. Ta sẽ chọn các điểm bắt đầu và chạm đất của quá trình nảy. Ta sẽ chọn 4 điểm: Điểm ảnh bắt đầu rơi xuống (X5), điểm ảnh chạm đất lần 1 (X4), điểm ảnh nảy lên cực đại lần 1 (X3), điểm ảnh chạm đất lần 2 (X2), điểm ảnh nảy lên cực đại lần 2 (X1) và điểm ảnh làm ảnh trùng với khung hình (X0).

- Ở đây, ta sẽ chọn 3 hàm parabol. Hàm 1 có gốc tại X5 và đi qua X4. Hàm 2 gốc tại X3, đi qua X4 và X2. Hàm 3 có gốc tại X1, đi qua X2 và X0. Như vậy, tâm hình sẽ di chuyển theo quỹ đạo của từng parabol (Với thứ tự từ X5->X0)



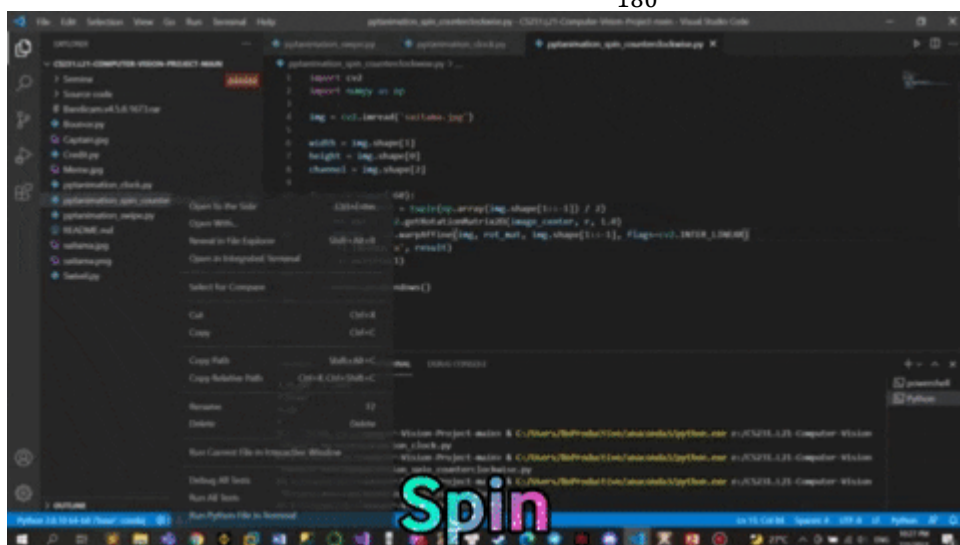
Hình 5.4. 5. Hiệu ứng Bounce

2.4.3. Spin

Đối với hiệu ứng Spin, Việc cần làm là xoay ảnh theo trung tâm ảnh 1° mỗi lần xoay và lặp lại 360 lần. Cứ mỗi lần xoay như vậy thì ta sẽ xuất ảnh ra 1 lần. Tập hợp những ảnh được hiển thị ra như vậy sẽ tạo thành hiệu ứng Spin. Để xoay 1° mỗi lần như vậy thì ta sử dụng affine transformation với ma trận xoay là:

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

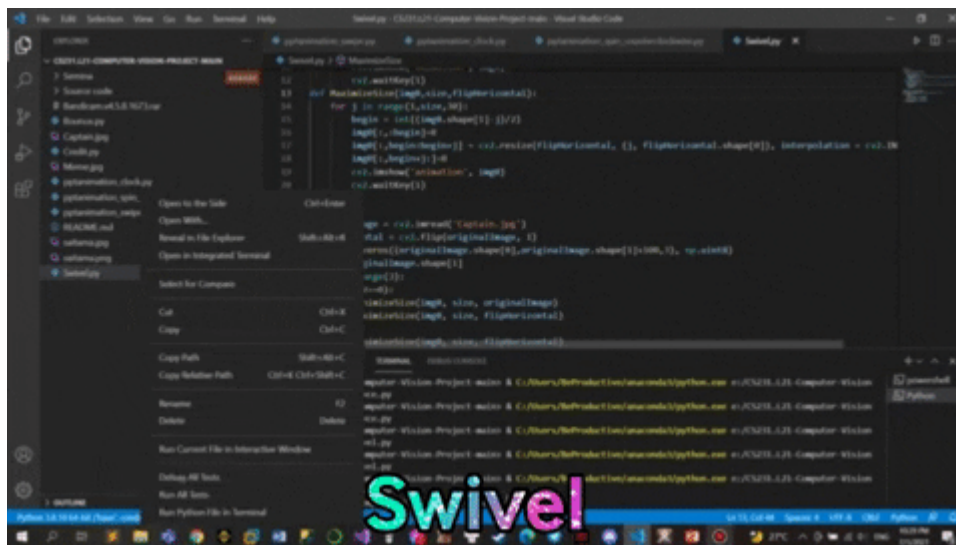
Trong đó θ là góc quay. Ở trường hợp này $\theta = 1^\circ = \frac{\pi}{180}$



Hình 5.4. 6. Hiệu ứng spin

2.4.4. Swivel

Thoạt nhìn thì hoạt ảnh này giống một bức ảnh xoay quanh một trục thẳng đứng ở giữa ảnh, nhưng sự việc thì đơn giản hơn nhiều. Ta cần có hai ảnh, một ảnh gốc và một ảnh đối xứng theo trục Oy. Bắt đầu bằng việc thu nhỏ dần chiều rộng của ảnh gốc và ảnh đối xứng cho đến khi nhỏ nhất. Sau đó hiển thị từng ảnh theo thứ tự ảnh gốc nhỏ dần, ảnh đối xứng to dần, ảnh đối xứng nhỏ dần rồi đến ảnh gốc to dần. Như vậy là có một vòng quay quanh trục của ảnh. Có thể lặp lại nhiều lần để tăng số vòng quay.



Hình 5.4. 7. Hiệu ứng Swivel

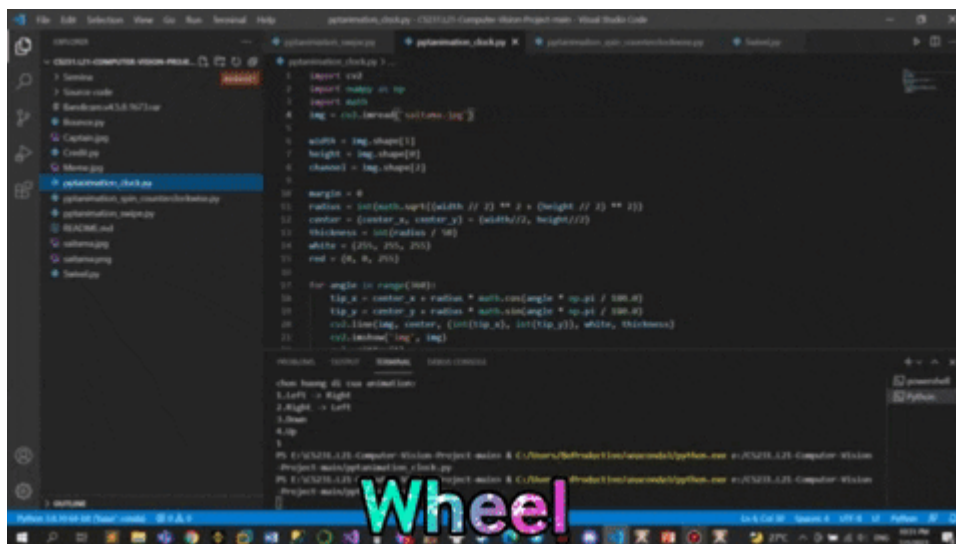
2.4.5. Wheel

Đối với hoạt ảnh này, có thể tưởng tượng ra một chiếc đồng hồ. Khi kim giây đi tới đâu thì ảnh sẽ trắng tới đó. Ta mô phỏng kim giây như một đường thẳng từ tâm cho đến rìa ngoài. Vị trí tâm ảnh có thể tính được bằng chiều dài chia hai và chiều rộng chia đôi. Còn vị trí đầu cây kim sẽ là (tip_x, tip_y) , sao cho:

$$tip_x = x + d * \cos \theta$$

$$tip_y = y + d * \sin \theta$$

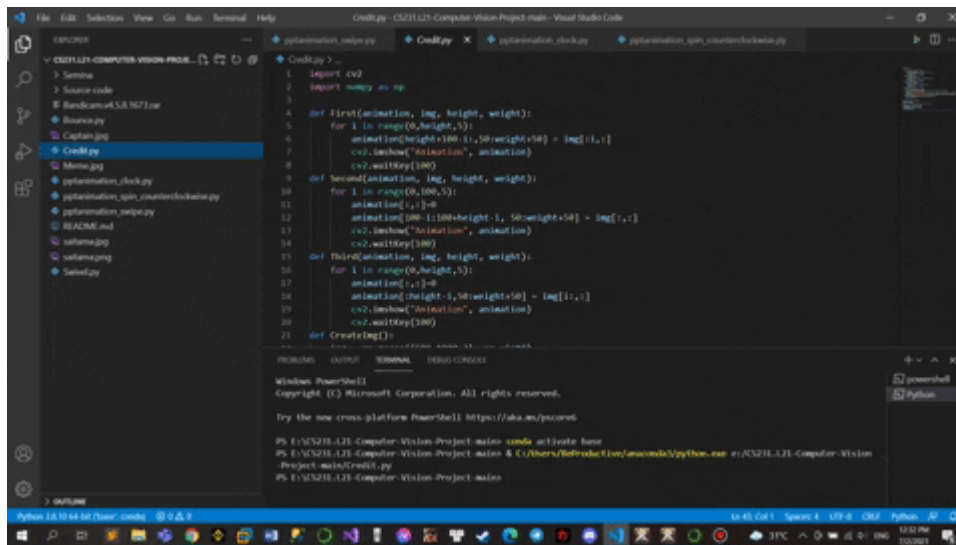
Trong đó: x, y là vị trí trung tâm ảnh, d là đường chéo ảnh chia đôi, θ là góc quay. Cứ mỗi vòng lặp thì tính điểm cuối kim giây sau khi quay góc 1° và vẽ đường thẳng từ tâm cho đến điểm đó. Sau đó hiển thị từng ảnh mỗi khi bị thay đổi ta sẽ được hoạt ảnh wheel.



Hình 5.4. 8. Hiệu ứng Wheel

2.4.6. Float in

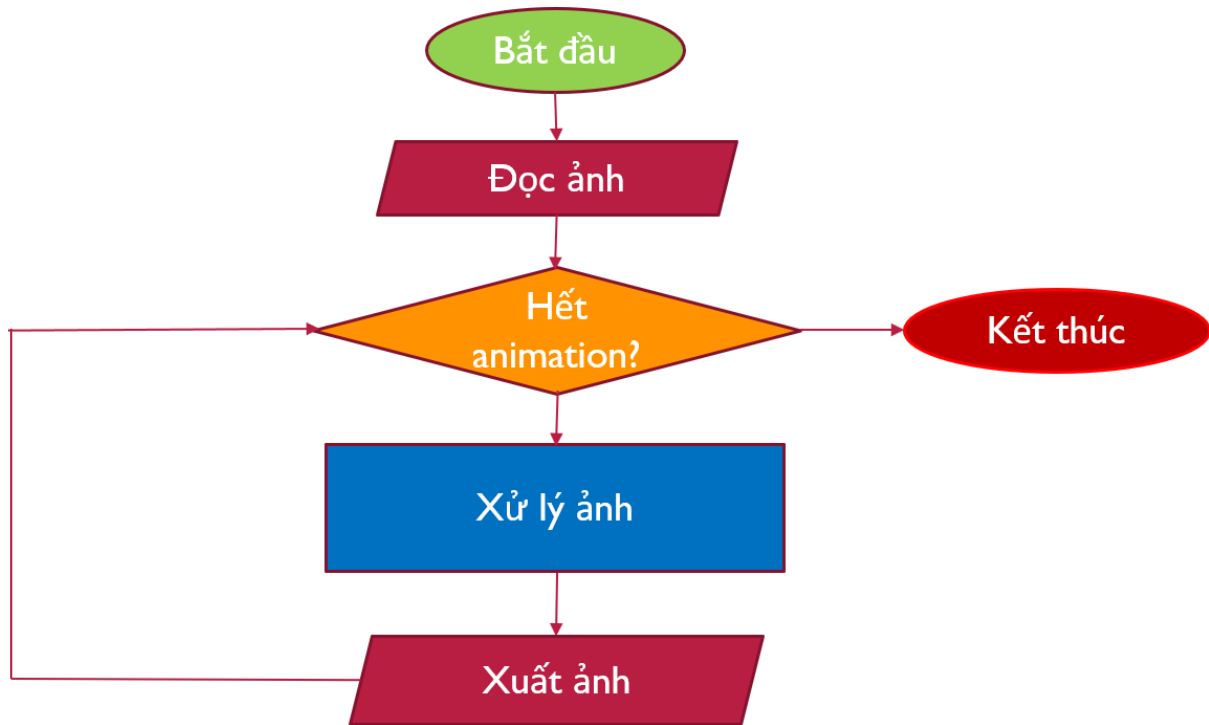
Với hoạt ảnh này, ta tạo một nền đen và một ảnh có chứa nội dung. Sau mỗi vòng lặp thì đổi các pixel ở nền đen bằng các pixel của ảnh, vị trí đổi thì tăng từ dưới lên. Sau mỗi lần lặp như vậy thì xuất ra màn hình sẽ tạo thành hoạt ảnh Float in.



Hình 5.4. 9. Hiệu ứng float in

2.5. Phương pháp thực hiện chung

- Trong Powerpoint có rất nhiều animation, nhưng hầu hết chúng đều đi qua các bước cơ bản như sau:



- Đầu tiên là đưa vào ảnh muốn có animation. Sau đó thực hiện các thay đổi với ảnh, mỗi lần thực hiện thay đổi thì xuất ảnh. Lặp lại bước trên cho đến khi hết hoạt ảnh. Tập hợp các ảnh được xuất ra chính là animation. Lưu ý là có thể thay vì xuất ảnh ta sẽ lưu ảnh lại, và kết hợp lại thành một gif hay một video để sử dụng.

- Cụ thể hơn đối với các animation dạng thoát ra như Swipe, Split, Shape, Wheel, Random bars thì có một số thay đổi với thuật toán như sau:

B1: <Chuyển hệ màu của ảnh sang HSV>. Bước này có thể có hoặc không tùy cách sử dụng.

B2: Duyệt qua ảnh theo thứ tự nhất định như từ trên xuống, từ dưới lên, theo chiều kim đồng hồ, ngẫu nhiên, ...

B3: Duyệt tới đâu thì thực hiện thay đổi ảnh tới đó. Các thay đổi có thể là:

- + giảm V và tăng S (nếu thực hiện bước 1)

- + thay thế pixel ảnh bằng màu trắng, đen, ...

B4: Chiều từng hình sau mỗi lần thực hiện thay đổi

- Đối với các animation có thay đổi hình dạng ảnh hay vị trí ảnh như Fly in, Swivel, Bounce, Float in, Spin, Teeter, ... thì sẽ có các bước làm hơi khác một chút:

B1: Chạy vòng lặp với số lần hữu hạn hay là lặp cho đến khi được xem là hết hoạt ảnh.

B2: Mỗi lần lặp thì thực hiện thay đổi ảnh, các thay đổi có thể là:

- + tăng giảm kích thước ảnh
- + thay đổi vị trí ảnh so với khung hình
- + sử dụng affine transformation

B3: Chiếu từng hình sau mỗi lần thực hiện thay đổi

- Ngoài ra còn có nhiều cách khác không bắt buộc phải thực hiện theo các bước trên. Nhưng các bước ở trên là những phương pháp để tạo thành animation một cách đơn giản.

2.6. Source code python và demo

- Link github source code python và video demo các animation:

<https://github.com/19522515/CS231.L21-Computer-Vision-Project>

Chương 3: CÀI ĐẶT VÀ KIỂM THỬ

3.1 Danh sách các animation

STT	Tên Animation
1	Spin
2	Wipe (Bao gồm 4 chức năng)
3	Bounce
4	Swivel
5	Wheel
6	Credit

3.2 Mức độ hoàn thành các animation

STT	Tên animation	Mức độ hoàn thành(%)	Ghi chú
1	Spin	100%	
2	Wipe	100%	
3	Bounce	100%	
4	Swivel	100%	
5	Wheel	100%	
6	Credit	100%	

Chương 4: NHẬN XÉT VÀ KẾT LUẬN

4.1 Nhận xét

4.1.1 Kết quả đạt được

- Nhóm đã tìm hiểu và hoàn thành được các animation với các yêu cầu được đưa ra.
- Thực hiện được yêu cầu và quy định đã đặt ra từ ban đầu.

4.1.2 Hạn chế

- Chưa thực hiện được tất cả các animation trong powerpoint.
- Animation có thể không giống y hệt trong powerpoint.

4.1.3 Kết luận

- Nhóm em đã hoàn thành đa số các yêu cầu mà đề án đưa ra và có bổ sung thêm các tính năng khác. Tuy nhiên, nhóm còn một số lỗi cần khắc phục như sau: giao diện animation chưa thật sự đẹp, vẫn còn các lỗi tiềm ẩn bên trong chưa được phát hiện.

4.2 Hướng phát triển

- Xây dựng thêm các powerpoint animation chưa cài đặt.
- Cải thiện giao diện, hiệu ứng đẹp hơn cho animation.
- Thiết kế app tự động chỉnh animations cho các slides mà người dùng đưa vào.

Chương 5: TÀI LIỆU THAM KHẢO

- [1] Slide Bài giảng trên course UIT , Microsoft Team
- [2] Giáo trình Nhập môn thị giác máy tính
- [3] https://vi.wikipedia.org/wiki/Microsoft_PowerPoint
- [4] <https://dacocach.com/powerpoint/>
- [5] <https://www.geeksforgeeks.org/python-opencv-affine-transformation/>
- [6] https://vi.wikipedia.org/wiki/Mô_hình_màu_RGB
- [7] <https://stackoverflow.com>
- [8] https://docs.opencv.org/3.4/d4/d61/tutorial_warp_affine.html
- [9] https://vi.wikipedia.org/wiki/Mô_hình_màu_CMYK
- [10] <https://vi.eyewated.com/mo-hinh-mau-hsv-la-gi/>
- [11] <https://vi.wikipedia.org/wiki/YCbCr>

Chương 6: BẢNG PHÂN CÔNG CÔNG VIỆC

STT	MSSV	Công việc được giao	Mức độ hoàn thành(%)
1	19521754	Viết code, thuyết trình seminar phân lý thuyết, hỗ trợ làm slide powerpoint và báo cáo đồ án	100%
2	19522515	Làm slide powerpoint, viết báo cáo đồ án, hỗ trợ viết code, soạn câu hỏi trả lời trong seminar	100%
3	19522102	Thuyết trình seminar phân demo, hỗ trợ viết code, hỗ trợ làm slide powerpoint và báo cáo đồ án	100%

□□ **Hết** □□