



COMPUTER GRAPHICS



Lưu Thị Hồng Ngọc
Hồ Thiên Phước

Table of Contents

Tuần 1: INTRODUCTION TO COMPUTER GRAPHICS (07/10/2021).....	4
Câu 1: CAD, CAM là gì?	4
Câu 2: Tại sao CAD và CAM là hai lĩnh vực đem lại doanh thu cao trong lĩnh vực đồ họa?	4
Câu 3: Nêu ba lĩnh vực sử dụng đhmt hiệu quả nhất ở VN?.....	4
Câu 4: Trong sản xuất, khi sử dụng máy in 3D để tạo ra sp có những thay đổi về nguyên lý cơ bản như thế nào so với sản xuất sp bình thường theo quy trình hiện nay?.....	4
Câu 5: Cho ví dụ một số phần mềm nào áp dụng vào trong giáo dục hiện nay có thể đem lại ứng dụng lớn (hiện nay hoặc trong tương lai có thể làm tăng chất lượng đào tạo)?	5
Câu 6: Sự khác biệt giữa cách học Đồ họa máy tính của những người IT và người không chuyên IT?	5
Câu 7: Tóm tắt nội dung của CMCN 4.0?	5
Câu 8: Tại sao IoT lại được quan trọng hóa? Tầm quan trọng của IoT?.....	5
Câu 9: Vì sao nói AI là linh hồn của chuyển đổi số của cuộc CMCN 4.0 làm thông minh hóa QTSX và QLXH ở diện rộng?	5
Câu 10: Thông minh là gì? Hệ thống như thế nào là thông minh? Giải thích vì sao “Thông minh là khả năng thích ứng với sự thay đổi”?	6
Câu 11: Hệ thống “thông minh” ở lĩnh vực Intelligence Vision System cần đạt những tiêu chí gì? ..	6
Câu 12: Nêu ví dụ hệ thống nào có Intelligence System mà performance càng dùng càng cao?	6
Câu 13: Phân biệt sự khác nhau giữa ba môn đồ họa máy tính, xử lý ảnh video và thị giác máy tính?.....	6
Câu 14: Cho ví dụ về hệ thống ứng dụng cả ba lĩnh vực trên.....	7
Câu 15: Cho ví dụ Intelligent Vision System có output đạt được nhiều cấp độ:	7
Tuần 2: INTRODUCTION TO COMPUTER GRAPHICS (CONT) (14/10/2021).....	8
Câu 1: Video thứ 1 nói về cái gì? Cái đó nó làm gì?	8
Câu 2: Video thứ 2 muốn nói gì ?	8
Câu 3: SLAM viết tắt của gì? Giải thích SLAM trong Visual SLAM?.....	8
Câu 4: Nhận dạng hành động để làm gì? Ở tầng nào? (Tầng 1: xử lý dữ liệu cơ bản, Tầng 2: tác vụ đơn, Tầng 3: ứng dụng)	9
Câu 5: Dự đoán gương mặt theo thời gian để làm gì? Xử lý ở tầng nào?.....	9
Câu 6: Nhận dạng y phục để làm gì?	9
Câu 7: Tại sao yêu cầu nhận dạng dựa trên khung xương?	9
Tuần 3: LINE + CIRCLE DRAWING (04/11/2021)	9
Câu 1: Nguyên lý vẽ đoạn thẳng là gì?	9

Câu 2: Vẽ đoạn thẳng với hệ số góc m tùy ý với phương trình $y=mx+b$	9
Câu 3: Tìm giải pháp vẽ đoạn thẳng?	9
Câu 4: Có thể giảm phức tạp từ $*float$ thành $\pm float$. Vậy có thể giảm từ $\pm float$ thành $\pm int$ không?	10
Câu 5: Cách khử m, b	10
Câu 6: Tính p_0	10
Câu 7: Vẽ đoạn thẳng từ 2 điểm (20,10) và (30,18)	11
Câu 8: Vẽ ngôi sao 5 cánh đều với r_1, r_2 cm cho trước và $r_1 > r_2$	12
Câu 9: Tính số viên gạch tròn $R=a$ cm, viên vườn hình ngôi sao 5 cánh đều biết r_1, r_2 cm	13
Câu 10: Vẽ tam giác 3 đcao, 3 trung tuyến, 3 phân giác, 3 trung trực (phương pháp, giải thuật) ..	13
Câu 11: Cho x tăng đều, tính y theo x	13
Câu 12: Vẽ cung tròn AB bán kính r	13
Câu 13: Tính số viên gạch $R=a$, viên hồ hình tròn R cm.....	14
Tuần 4: CIRCLE DRAWING (11/11/2021)	14
Câu 1: $f_{circle}(x,y) < 0$ khi nằm bên trong đường tròn và > 0 sẽ nằm ngoài đường tròn, đúng hay sai?	14
Câu 2: Tính p_0	14
Câu 3: Chứng minh nếu $r: int$ thì có thể thay $p_0=5/4-r$ thành $p'_0=1-r$ (hint: chứng minh $p'_0 \geq 0 \Leftrightarrow p_0 \geq 0$).....	14
Câu 4: (0;10)	14
Câu 5: Chỉ ra ý nghĩa hình học của góc θ (nó là góc nào trong Elip)?	16
Câu 6: Vẽ Elip $r_x=8, r_y=6$	16
Tuần 5: BEZIER (18/11/2021)	18
Câu 1:Viết pt đường cong Bezier.....	18
Câu 2: So sánh các điều kiện biên với tính chất của đường cong Bezier	18
Câu 3: Sai phân. Tính $x_k = x(uk)$	18
Tuần 6 BIEZIER (CONT) AND FILLING (25/11/2021)	19
Câu 1: Vẽ đường cong Biezer theo phương pháp của De Casteljau. Kiểm chứng method 2 ông có giống nhau(trùng) không (ans: giống)	19
Câu 2: Tại sao không tô màu tràn ra được ?(nguyên lí Boundary Fill).....	19
Câu 3: Rã đệ qui Boundary Fill	19
Câu 4: Tiêu chí để tự động xét điểm cực trị (loại bỏ cạnh //Ox) và k cực trị (ngắn cạnh)	22
Câu 5: Tính giao điểm dòng quét vs các cạnh (hint: công thức qui nạp tính giao điểm dòng vét với 1 cạnh)	22
Câu 6: Tại sao phải sort giao điểm theo thứ tự tăng dần trong Scan-line Polygon Fill method	22

Tuần 7 FILLING (CONT) AND CLIPPING : 02/12/2021	23
Câu 1	23
Câu 2: Tìm biểu thức toán thể hiện các trường hợp, mối quan hệ vị trí (nằm ngoài, nằm trong, cắt..) Cohen Sutherland line clipping 8.1.2	29
Câu 3: Làm sao cho máy tính biết p9p10 nằm ngoài	29
Câu 4: Hàm encode để biến điểm thành mã vùng	30
Câu 5: inside().....	30
Câu 6: outside()	30
Tuần 8 GEOMETRIC TRANSFORMATION (09/12/2021)	30
Câu 1: Làm sao để tính ra công thức 2D Rotation.....	30
Câu 2: Ma trận biến đổi co giãn scale tâm C.....	31
Câu 3: Shear Trượt tương đối so với trục Ox	31
Câu 4: Quay quanh trục //Ox,Oy,Oz qua tâm C 9.4.2	32
Câu 5: Giải thuật auto phát sinh tập đỉnh nằm trên kinh tuyến vĩ tuyến của quả cầu theo phương trình tọa độ cầu $x^2 + y^2 + z^2 = r^2$ Quadratic Surface	32
Câu 6: 2 loop kết nối các đỉnh 10.1.2	34
Câu 7: Viết cấu trúc dữ liệu để quản lí bảng đỉnh, mặt, cạnh	35
Tuần 9 PROJECTION TRANSFORM (16/12/2021).....	36
Câu 1: Viết lại d 16:00 chiếu Perspective	36
Câu 2: Vẽ/mô phỏng quả cầu tâm C có R với mặt phẳng tùy ý	37
Câu 3: Xác định matrix chiếu gồm những ma trận con nào? với mp chiếu tùy ý, vector chiếu V Parallel (hint: Quay n trùng Oz).....	37
Câu 4: Tính depth.....	37
Câu 5: Scanline method trong khử mặt khuất: giữ cái nào, bỏ cái nào?	37
Framework của các nhóm	38

Tuần 1: INTRODUCTION TO COMPUTER GRAPHICS (07/10/2021)

Câu 1: CAD, CAM là gì?

→CAD: computer aided design

CAM: computer aided manufacturing

-Là kỹ thuật thiết kế và sản xuất có sự trợ giúp của máy tính như gia công, vẽ

-Tạo ra những sản phẩm chính xác, độ bền, tính thẩm mỹ tuyệt vời

CAD:

- Tạo ra các bề mặt
- Sử dụng các hình dạng hình học để xây dựng một mô hình

CAM:

- Thực hành qui trình sản xuất, gia công giả lập trên máy tính
- Nhập liệu và tính toán các đường cắt, cắt gọt loại bỏ vật liệu dư thừa
- Tối ưu hóa đường chạy dao

Câu 2: Tại sao CAD và CAM là hai lĩnh vực đem lại doanh thu cao trong lĩnh vực đồ họa?

→CAD: hầu hết các sp trong các lĩnh vực đều cần có các bản thiết kế mô hình sp, trong sx thì không thể nào vẽ tay hết đc ☹ dùng phần mềm đồ họa bằng máy ☹ CAD là rất quan trọng

CAM: lấy dữ liệu bản vẽ từ CAD chuyển hóa thành ngôn ngữ mã lập trình → điều khiển các máy móc để tạo ra những sp.

Câu 3: Nêu ba lĩnh vực sử dụng đhmt hiệu quả nhất ở VN?

→Kiến trúc xây dựng: tiết kiệm thời gian vẽ, chỉnh sửa → tăng thời gian thiết kế → đem lại năng suất cao.

Cơ khí.

Game.

Câu 4: Trong sản xuất, khi sử dụng máy in 3D để tạo ra sp có những thay đổi về nguyên lý cơ bản như thế nào so với sản xuất sp bình thường theo quy trình hiện nay?

→3D trong quy trình sản xuất: → máy làm hết quá trình sản xuất, gia công mà ko cần sự giám sát, chỉnh sửa của con người

In 3D sản xuất trực tiếp ra sp mà ko cần qua khuôn đúc hoặc các khâu trung gian khác.

- Về nhân công: giảm nhân công, chi phí thuê thợ lành nghề
- Về thời gian sản xuất: sản xuất nhanh chóng.
- Về năng suất sản xuất: chất lượng ổn định, đồng loạt.

- Về hiệu quả và tính ứng dụng linh hoạt: ...

Câu 5: Cho ví dụ một số phần mềm nào áp dụng vào trong giáo dục hiện nay có thể đem lại ứng dụng lớn (hiện nay hoặc trong tương lai có thể làm tăng chất lượng đào tạo)?

→ Các phần mềm hỗ trợ vẽ hình học phẳng, hình học không gian, đồ thị số, ...

hoặc các phần mềm ứng dụng cho môn lý hóa sinh.

Câu 6: Sự khác biệt giữa cách học Đồ họa máy tính của những người IT và người không chuyên IT?

→ Cách học của người IT:

Tìm hiểu sâu về nguyên lý, cách thức hoạt động của đồ họa → tìm ra pp → xây dựng thuật toán → làm ra phần mềm đồ họa phục vụ cho những ng cần sử dụng các phần mềm đồ họa để thiết kế, ...

⇒ Người IT cần phải nắm rõ nguyên lý đồ họa để có thể xây dựng thuật toán thích hợp và thiết kế ra phần mềm đồ họa cho phần lớn người cần sử dụng để thiết kế sp, trong khi đó thì những người không chuyên chỉ cần nắm rõ cách sử dụng công cụ để thiết kế ra sp mà không cần hiểu sâu vào nguyên lý hđ của phần mềm đồ họa.

Câu 7: Tóm tắt nội dung của CMCN 4.0?

→ Ở cuộc CMCN 4.0, sử dụng và ứng dụng những thiết bị điện tử vào cuộc sống (ví dụ ra ...)

→ CMCN 4.0 cũng sử dụng thiết bị đt nhưng ở mức độ cao hơn. ⇒ chuyển từ giai đoạn sử dụng máy móc sang giai đoạn sử dụng máy móc tự động, chuyển đổi số.

AI, IoT, Robotic, Big Data, ... là những nhân tố quan trọng trong CMCN 4.0.

Cốt lõi của CMCN 4.0: thông minh hóa quá trình sản xuất và quản lý xã hội ở diện rộng.

Dựa trên nền tảng chuyển đổi số → pp chính: sử dụng AI.

Câu 8: Tại sao IoT lại được quan trọng hóa? Tầm quan trọng của IoT?

→ Tính đồng bộ, tự động hóa mà không cần có sự can thiệp của con người, lấy và lưu trữ thông tin tự động, có khả năng chia sẻ, liên kết thông tin giữa vật với vật thông qua mạng internet. → quản lý đô thị, sản xuất thông minh, phản ánh được các thông tin một cách nhanh chóng, kịp thời.

Câu 9: Vì sao nói AI là linh hồn của chuyển đổi số của cuộc CMCN 4.0 làm thông minh hóa QTSX và QLXH ở diện rộng?

→ Cuộc cách mạng công nghệ lần thứ tư này đang làm thay đổi cách thức sản xuất, chế tạo. Trong các “nhà máy thông minh”, các máy móc được kết nối Internet và liên kết với nhau qua một hệ thống có thể tự hình dung toàn bộ quy trình sản xuất rồi đưa ra quyết định sẽ thay thế dần các dây chuyền sản xuất trước đây. Nhờ khả năng kết nối của hàng tỷ người trên thế giới thông qua các thiết bị di động và khả năng tiếp cận được với cơ sở dữ liệu lớn, những tính năng xử lý thông tin sẽ được nhân lên bởi những đột phá công nghệ trong các lĩnh vực như trí tuệ nhân tạo, công nghệ người máy, Internet kết nối vạn vật, xe tự lái, công nghệ in 3 chiều, công nghệ nano, công nghệ sinh học, khoa học vật liệu, lưu trữ năng lượng và tính toán lượng tử.

Qui mô và tốc độ phát triển - Chưa có tiền lệ trong lịch sử nhân loại. Tốc độ phát triển của những đột phá trong cách mạng công nghiệp lần thứ tư này là không có tiền lệ trong lịch sử. Nếu như các cuộc cách mạng công nghiệp trước đây diễn ra với tốc độ theo cấp số cộng (hay tuyến tính) thì tốc độ phát triển của cách mạng công nghiệp lần thứ tư này là theo cấp số nhân. Thời gian từ khi các ý tưởng về công nghệ và đổi mới sáng tạo được phôi thai, hiện thực hóa các ý tưởng đó trong các phòng thí nghiệm và thương mại hóa ở qui mô lớn các sản phẩm và qui trình mới được tạo ra trên phạm vi toàn cầu được rút ngắn đáng kể. Những đột phá công nghệ diễn ra trong nhiều lĩnh vực như kể trên với tốc độ rất nhanh và tương tác thúc đẩy nhau đang tạo ra một thế giới được số hóa, tự động hóa và ngày càng trở nên hiệu quả và thông minh hơn.

Câu 10: Thông minh là gì? Hệ thống như thế nào là thông minh? Giải thích vì sao “Thông minh là khả năng thích ứng với sự thay đổi”?

→ Hệ thống áp dụng và ứng dụng cho nhiều tình huống và giải quyết được nhiều vấn đề, có khả năng thích ứng tốt với những thay đổi...

Câu 11: Hệ thống “thông minh” ở lĩnh vực Intelligence Vision System cần đạt những tiêu chí gì?

→ Có thể ứng dụng vào nhiều lĩnh vực khác nhau, có thể nhận biết, học hỏi để thích ứng với nhiều tình huống khác nhau có thể xảy ra.

Câu 12: Nêu ví dụ hệ thống nào có Intelligence System mà performance càng dùng càng cao?

→ Hệ thống thông minh tự đề cử, chọn lọc thông tin mà người dùng đang quan tâm hoặc thường xem để hiển thị lên nhiều hơn hoặc tự động đề cử đến người dùng. Ví dụ như các trang mạng xã hội hoặc video được đề cử trên facebook, ...

Câu 13: Phân biệt sự khác nhau giữa ba môn đồ họa máy tính, xử lý ảnh video và thị giác máy tính?

→

	Computer Graphics (Đồ họa máy tính)	Digital Image & Video Processing (Xử lý ảnh số và video)	Computer Vision (Thị giác máy tính)
Mục Tiêu	Mô phỏng thế giới thực, ảo. Ví dụ như mô phỏng hoạt cảnh, mô hình, đổ bóng, mô phỏng chuyển động, ...	Tăng cường chất lượng hình ảnh, video đầu ra. Ví dụ như nén ảnh, giảm nhiễu, tìm cạnh, phân vùng, khôi phục ảnh, điều chỉnh độ tương phản.	Tăng cường khả năng hiểu ảnh. Ví dụ như truy vết, phát hiện, phân loại, nhận dạng, phân vùng ngữ nghĩa

Input	2D/ 3D points + entity name	Hình ảnh, video chưa qua xử lý (ảnh, video gốc)	Hình ảnh, video (raw)
Output	2D/ 3D shape, + texture, view, lighting, animation	Hình ảnh, video đã qua xử lý, tăng cường chất lượng.	Semantic meaning, 3D environment reconstruction

Câu 14: Cho ví dụ về hệ thống ứng dụng cả ba lĩnh vực trên.

→

DHMT	TGMT	XLA
Thiết kế hệ thống cơ khí, hệ thống cấp quang	Hệ thống tái tạo hình ảnh trong phục hồi ảnh hoặc phim đen trắng. Hoặc trong xe tự lái, tái tạo hình ảnh có thể được sử dụng để nhìn xa hơn các vật cản nhỏ, chẳng hạn như một biển chỉ dẫn giữa xe và người đi bộ đang được theo dõi.	Hệ thống chụp ảnh và tín hiệu ảnh, hệ thống số hóa ảnh: Lấy mẫu, Lượng tử hoá, hệ thống nhận dạng, nén ảnh, hệ thống lọc trên miền tần số, hệ thống khôi phục ảnh

Câu 15: Cho ví dụ Intelligent Vision System có output đạt được nhiều cấp độ:

→

System name		Person tracking
Goal		-Nhận dạng đặc điểm, hành vi của con người thông qua camera ghi hình -Nhận dạng quang cảnh -Truy vết đối tượng
Input		Video quay được từ camera an ninh
Output	Data	Tất cả bản ghi từ camera an ninh qua các lần mở (dữ liệu chưa được xử lý)
	Information	Thông tin, hành vi của người ở nơi công cộng
	Knowledge	Nhận biết trong quang cảnh có người

	Intelligent	-Nhận biết hành vi ‘đang đứng’ hay ‘đứng và đưa tay lên nghe điện thoại’ -Nhận biết thông tin người trong khung hình như tuổi tác, giới tính, tóc ngắn hay dài, màu tóc, có mang đồ phụ tùng trên người hay không, màu sắc của nửa người trên và nửa người dưới, toạ độ để phân biệt đứng yên hay di chuyển, màu sắc giày dép,.. -Nhận biết được quang cảnh: lối vào lối ra, nhà ga, ngã tư,..
	Wisdom	Dự đoán hành vi của đối tượng, quang cảnh như: địa điểm đi kế tiếp, cảnh báo hành vi nguy hiểm,..

Tuần 2: INTRODUCTION TO COMPUTER GRAPHICS (CONT) (14/10/2021)

Câu 1: Video thứ 1 nói về cái gì? Cái đó nó làm gì?

→ Con robot tự hành được gắn camera quan sát chuyển động, có thể theo sát và đi theo đối tượng, nhận biết và tái tạo lại môi trường xung quanh trong phạm vi indoor.

Câu 2: Video thứ 2 muốn nói gì ?

→ Drone có hình dạng cây dù, theo vết được người dùng qua camera theo dõi.

-Sử dụng chủ đề AR trong lĩnh vực ảo ảnh quang học

-Với chế độ thiết lập trong ứng dụng, chiếc ô có thể giữ khoảng cách nhất định đối với đối tượng động

Câu 3: SLAM viết tắt của gì? Giải thích SLAM trong Visual SLAM?

→ SLAM - Simultaneous Localization And Mapping - là vấn đề phổ biến với một số ứng dụng liên quan đến điều khiển tự động, nghiên cứu về thiết bị tự hành ví dụ như UAV, drone, flycam, humanoid, robot.. Dùng camera tiếp nhận thông tin thị giác để định hướng đường đi.

Cụ thể, hệ thống sử dụng thông tin ảnh thu được từ camera để tái tạo môi trường bên ngoài bằng cách đưa thông tin môi trường vào một map (2D hoặc 3D), từ đó thiết bị (robot, camera, xe) có thể định vị (localization) đang ở đâu, trạng thái, tư thế của nó trong map để tự động thiết lập đường đi (path planning) trong môi trường hiện tại.

Điều khiển tự động thiết bị robot chia làm 3 vấn đề chính:

- Định vị (localization)- cần xác định vị trí hiện tại của robot dựa vào bản đồ tái tạo.
- Tái tạo môi trường (mapping).
- Hoạch định đường đi (path planning).

Tái tạo bản đồ: cần xác định vị trí của đối tượng trong bản đồ, để xây dựng bản đồ chính xác nhất, ít sai số.

Câu 4: Nhận dạng hành động để làm gì? Ở tầng nào? (Tầng 1: xử lý dữ liệu cơ bản, Tầng 2: tác vụ đơn, Tầng 3: ứng dụng)

→ Dự đoán các hành động tiêu cực → giám sát an ninh, giao thông

Tăng khả năng tương tác giữa người với máy, máy với máy,...

Câu 5: Dự đoán gương mặt theo thời gian để làm gì? Xử lý ở tầng nào?

→ Tạo và thu thập dữ liệu phong phú về mặt người. Phục vụ cho việc truy vết con người theo thời gian (ví dụ như tìm người thất lạc, truy tìm tội phạm, ...)

Câu 6: Nhận dạng y phục để làm gì?

→ Nhận dạng y phục nhằm:

- Giúp ích nắm bắt thông tin về xu hướng thời trang ở phạm vi rất rộng.
- Nắm bắt thị yếu, thị trường.
- Truy vết đối tượng bằng khuôn mặt gặp khó khăn thì nhận dạng y phục sẽ là giải pháp thay thế.

Câu 7: Tại sao yêu cầu nhận dạng dựa trên khung xương?

→ Vì người chỉ chiếm diện tích nhất định trong một background và đôi khi background sẽ bị nhiễu.

Tuần 3: LINE + CIRCLE DRAWING (04/11/2021)

Câu 1: Nguyên lý vẽ đoạn thẳng là gì?

Đối số đầu vào: Tọa độ của 2 đầu mút

Ảnh số là đoạn thẳng- tập pixel được vẽ dựa trên phương trình đường thẳng, trên lưới tọa độ nguyên.

Yêu cầu bài toán:

- Các pixel phải đạt được giá trị xấp xỉ tốt nhất.
- Các pixel phải thỏa được tính liên tục trong lân cận 8 về mặt không gian.
- Độ phức tạp tính toán phải thấp.

Câu 2: Vẽ đoạn thẳng với hệ số góc m tùy ý với phương trình $y=mx+b$

$$0 < m \leq 1 \Rightarrow 0 < \frac{\Delta y}{\Delta x} \leq 1$$

$$\Rightarrow \begin{cases} \Delta x = 1 \Rightarrow y_x \\ \Delta y \leq 1 (nhân) \\ \Delta y = 1 \Rightarrow x_y \\ \Delta x \geq 1 (loại) \end{cases}$$

Câu 3: Tìm giải pháp vẽ đoạn thẳng?

Phương pháp:

$$y = mx + b$$

$$\Delta y = m \cdot \Delta x \Rightarrow \begin{cases} 0 < m \leq 1 : \Delta x = 1, 0 < \Delta y \leq 1 \Rightarrow \begin{cases} x_{k+1} - x_k = 1 \\ y_{k+1} - y_k = m \end{cases} \Rightarrow \begin{cases} x_{k+1} = x_k + 1 \\ y_{k+1} = y_k + m \end{cases} \\ m > 1 : \Delta y = 1 \Rightarrow |\Delta x| \leq 1 \Rightarrow \begin{cases} y_{k+1} - y_k = 1 \\ x_{k+1} - x_k = \frac{1}{m} \end{cases} \Rightarrow \begin{cases} y_{k+1} = y_k + 1 \\ x_{k+1} = x_k + \frac{1}{m} \end{cases} \end{cases}$$

Giải thuật:

```
void LineDDA(int xa,int ya,int xb,int yb){
    int dx=xb-xa,dy=yb-ya,step,k;
    float x=xa,y=ya,xIncr,yIncr;

    if(abs(dx)>abs(dy)) step=abs(dx);
    else step=abs(dy);

    xIncr=dx/step*1.0;
    yIncr=dy/step*1.0;

    setpixel(round(x),round(y));
    for(int i=0;i<step;i++){
        x+=xIncr;
        y+=yIncr;
        setpixel(round(x),round(y));
    }
}
```

Câu 4: Có thể giảm phức tạp từ *float thành ± float. Vậy có thể giảm từ ± float thành ±int không?

→ Không. Vì nếu làm như vậy thì giải thuật sẽ không chạy được do vòng lặp xảy ra lỗi.

Câu 5: Cách khử m,b

$$m = \frac{\Delta y}{\Delta x} \Rightarrow p_k = \Delta x(d_2 - d_1) = 2\Delta yx_k - 2\Delta xy_k + C \text{ (với } C = 2\Delta y + \Delta x(2b - 1))$$

$p_k < 0 \Rightarrow$ Chọn y_{k+1} là điểm cận dưới $\Rightarrow y_{k+1} = y_k$

$p_k \geq 0 \Rightarrow$ Chọn y_{k+1} là điểm cận trên

Qui nạp

$$\begin{aligned} p_{k+1} &= 2\Delta yx_{k+1} - 2\Delta xy_{k+1} + C \\ \Rightarrow p_{k+1} - p_k &= 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_{k+1} - y_k) \\ \Rightarrow p_{k+1} &= p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k) \end{aligned}$$

Câu 6: Tính p_0

$$p_0 = 2\Delta yx_0 - 2\Delta xy_0 + 2\Delta y + \Delta x(2b - 1) (*)$$

Vì $y_0 = \frac{\Delta y}{\Delta x} x_0 + b$ (phương trình đường thẳng). Thay vào (*):

$$\Rightarrow p_0 = 2\Delta y x_0 - 2\Delta x \left(\frac{\Delta y}{\Delta x} x_0 + b \right) + 2\Delta y + \Delta x(2b - 1)$$

$$\Rightarrow p_0 = 2\Delta y x_0 - 2\Delta y x_0 - 2\Delta x b + 2\Delta y + \Delta x(2b - 1)$$

$$\Rightarrow p_0 = 2\Delta y - \Delta x$$

Câu 7: Vẽ đoạn thẳng từ 2 điểm (20,10) và (30,18)

$$\Delta y = 18 - 10 = 8$$

$$\Delta x = 30 - 20 = 10$$

Ktra tham số m:

$$\frac{\Delta y}{\Delta x} = \frac{8}{10} < 1$$

Tính các thông số cơ bản:

$$\Rightarrow 2\Delta y = 16$$

$$\Rightarrow 2\Delta y - 2\Delta x = 14$$

$$\Rightarrow p_0 = 2\Delta y - \Delta x = 16 - 10$$

$p_k < 0$, vẽ điểm kế tiếp tại x_{k+1}, y_k với $p_{k+1} = p_k + 2\Delta y$

$p_k \geq 0$, vẽ điểm kế tiếp tại x_{k+1}, y_{k+1} với $p_{k+1} = p_k + 2\Delta y - 2\Delta x$

$$(x_0, y_0) = (20, 10)$$

Lập bảng tọa độ các pixel

k	p_k	x_{k+1}	y_{k+1}
0	6	21	11
1	2	22	12
2	-2	23	12
3	14	24	13
4	10	25	14
5	6	26	15
6	2	27	16
7	-2	28	16

8	14	29	17
9	10	30	18

18											x
17										x	
16								x	x		
15							x				
14						x					
13					x						
12			x	x							
11		x									
10	x										
	20	21	22	23	24	25	26	27	28	29	30

Câu 8: Vẽ ngôi sao 5 cánh đều với r_1, r_2 cm cho trước và $r_1 > r_2$

Nguyên lí:

input: r_1, r_2

output: ngôi sao 5 cánh tạo bởi 5 đỉnh và những cạnh là đường thẳng

Phương pháp:

Giả sử tâm ngôi sao là góc tọa độ O, cạnh a là trục Ox, góc $\alpha = \angle P_2OP_1 = \frac{360^\circ}{5} = 72^\circ$

Bước 1: Tính tọa độ các đỉnh

$$P_1(r_2, 0)$$

$$P_2(r_2 \cos 72^\circ, r_2 \sin 72^\circ)$$

$$P_3(r_2 \cos 144^\circ, r_2 \sin 144^\circ)$$

$$P_4(r_2 \cos 216^\circ, r_2 \sin 216^\circ)$$

$$P_5(r_2 \cos 288^\circ, r_2 \sin 288^\circ)$$

$$Q_1(r_1 \cos 36^\circ, r_1 \sin 36^\circ)$$

$$Q_2(r_1 \cos 108^\circ, r_1 \sin 108^\circ)$$

$$Q_3(r_1 \cos 180^\circ, r_1 \sin 180^\circ)$$

$$Q_4(r_1 \cos 252^\circ, r_1 \sin 252^\circ)$$

$$Q_5(r_1 \cos 324, r_1 \sin 324)$$

Bước 2: Dùng pp vẽ đoạn thẳng nối các đỉnh.

Câu 9: Tính số viên gạch tròn $R=a$ cm, viên vườn hình ngôi sao 5 cánh đều biết r_1, r_2 cm

Phương pháp

Giả sử 1 pixel tương ứng với 1 viên gạch tròn có $R = a$ cm \rightarrow Áp dụng cách vẽ ngôi sao 5 cánh đều ở câu 8 với $r'_1 = 2 * a * r_1$, $r'_2 = 2 * a * r_2$. Trong lúc vẽ đoạn thẳng nối các đỉnh, dùng biến đếm để tính số pixel. Sau đó lấy biến đếm trừ ra số đỉnh bị đếm lặp.

Câu 10: Vẽ tam giác 3 đcao, 3 trung tuyến, 3 phân giác, 3 trung trực (phương pháp, giải thuật)

Không

Câu 11: Cho x tăng đều, tính y theo x

$$(C): x^2 + y^2 = r^2$$

$$\Rightarrow 2x dx + 2y dy = 0$$

$$\left| \frac{dy}{dx} \right| = \left| -\frac{x}{y} \right| \leq 1$$

Do hệ số góc tiếp tuyến < 1 nên chỉ có thể tăng đều theo x và tính y theo x , nếu làm ngược lại thì cung tròn sẽ bị hở

Câu 12: Vẽ cung tròn AB bán kính r

Bước 1: Tính các thông số cơ bản

Lấy điểm khởi nguyên để tính tiếp cho bảng tọa độ

$$x_0 = x_A$$

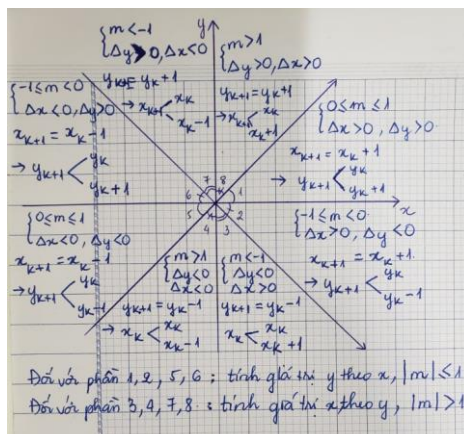
$$y_0 = r$$

$$\Rightarrow 2x_0 = 2x_A$$

$$\Rightarrow 2y_0 = 2r$$

$$p_0 = 1 - r$$

8 trường hợp trong cung tròn



Bước 2: Công thức tổng quát để lập bảng tọa độ các điểm

Nếu $p_k < 0 \Rightarrow p_{k+1} = p_k + 2 \cdot x_{k+1} + 1$ và $2x_{k+1} = 2x_k + 2$ và $y_{k+1} = y_k$

Nếu $p_k \geq 0 \Rightarrow p_{k+1} = p_k + 2 \cdot x_{k+1} - 2y_{k+1} + 1$ và $2x_{k+1} = 2x_k + 2$ và $2y_{k+1} = 2y_k - 2$

Điều kiện dừng: $(x_k, y_k) = (x_B, y_B)$

Câu 13: Tính số viên gạch $R=a$, viền hồ hình tròn R cm

Giả sử 1 pixel trong ô lưới tọa độ tương đương với kích thước viên gạch có $R = a \rightarrow$ Áp dụng phương pháp vẽ đường tròn với bán kính mới $R' = 2 \cdot a \cdot R$. Trong quá trình vẽ, dùng thêm biến đếm để xác định số pixel, sau đó trừ đi cho số pixel bị đếm lặp.

Tuần 4: CIRCLE DRAWING (11/11/2021)

Câu 1: $fcircle(x,y) < 0$ khi nằm bên trong đường tròn và > 0 sẽ nằm ngoài đường tròn, đúng hay sai?

Đúng. Vì đối với 1 điểm A nằm ngoài đường tròn, khoảng cách từ tâm O đến A sẽ lớn hơn bán kính $\rightarrow fcircle > 0$ và ngược lại.

Câu 2: Tính p_0

Với $x_0=0, y_0=r$

$$\Rightarrow p_0 = (0+1)^2 + (r-\frac{1}{2})^2 - r^2 = 5/4 - r$$

Câu 3: Chứng minh nếu $r: \text{int}$ thì có thể thay $p_0=5/4-r$ thành $p'_0=1-r$ (hint: chứng minh $p'_0 \geq 0 \Leftrightarrow p_0 \geq 0$)

$p_0=5/4-r$ (chứng minh trên)

$$p_0 = 1 + \frac{1}{4} - r$$

$$\Rightarrow p'_0 = p_0 - \frac{1}{4}$$

$$\text{Nếu } p'_0 \geq 0 \Rightarrow p_0 \geq \frac{1}{4} \Rightarrow p_0 \geq 0$$

Câu 4: (O;10)

Bước 1: Tính các thông số cơ bản

Lấy điểm khởi nguyên để tính tiếp cho bảng tọa độ

$$x_0=0$$

$$y_0=r=10$$

$$\Rightarrow 2x_0=0$$

$$\Rightarrow 2y_0=20$$

Bước 2: Lập bảng tọa độ các điểm trên góc $\frac{\pi}{8}$

Nếu $p_k < 0 \Rightarrow p_{k+1} = p_k + 2.x_{k+1} + 1$ và $2x_{k+1} = 2x_k + 2$ và $y_{k+1} = y_k$

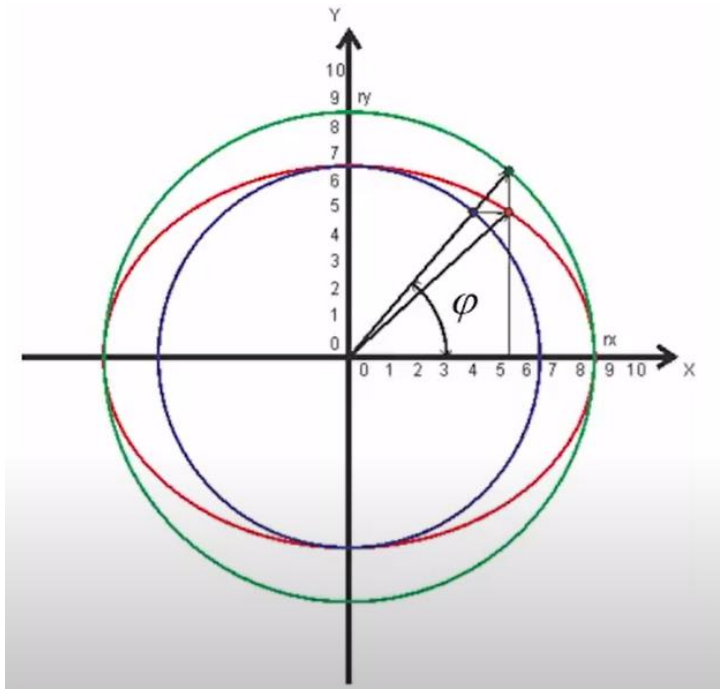
Nếu $p_k \geq 0 \Rightarrow p_{k+1} = p_k + 2.x_{k+1} - 2y_{k+1} + 1$ và $2x_{k+1} = 2x_k + 2$ và $2y_{k+1} = 2y_k - 2$

Điều kiện dừng: $x_{k+1} \geq y_{k+1}$

k	p_k	x_{k+1}	y_{k+1}	$2x_{k+1}$	$2y_{k+1}$
0	-9	1	10	2	20
1	-6	2	10	4	20
2	-1	3	10	6	20
3	6	4	9	8	18
4	-3	5	9	10	18
5	8	6	8	12	16
6	5	7	7	14	14

10	X	X	X	X							
9					X	X					
8							X				
7								X			
6									X		
5										X	
4										X	
3											X
2											X
1											X
0											X
	0	1	2	3	4	5	6	7	8	9	10

Câu 5: Chỉ ra ý nghĩa hình học của góc theta(nó là góc nào trong Elip)?



Câu 6: Vẽ Elip $r_x=8, r_y=6$

Bước 1: Tính các thông số cơ bản

$x_0=0$

$y_0=r_y=6$

$$2r_x^2 = 2 \cdot 8^2$$

$$2r_y^2 = 2 \cdot 6^2$$

$$p_{1,0} = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2 = -332$$

Bước 2: Lập bảng tọa độ các điểm trên góc $\frac{\pi}{8}$

Vùng 1

Nếu $p_{1,k} < 0 \Rightarrow p_{1,k+1} = p_{1,k} + 2 \cdot r_y^2 \cdot x_{k+1} + r_y^2$ và $x_{k+1} = x_k + 1$ và $y_{k+1} = y_k$

Nếu $p_{1,k} \geq 0 \Rightarrow p_{1,k+1} = p_{1,k} + 2 \cdot r_y^2 \cdot x_{k+1} - 2 \cdot r_x^2 \cdot y_{k+1} + r_y^2$ và $x_{k+1} = x_k + 1$ và $y_{k+1} = y_k - 1$

Điều kiện dừng: $2r_y^2 \geq 2r_x^2$

k	$p_{1,k}$	x_{k+1}	y_{k+1}	$2 \cdot r_y^2 \cdot x_{k+1}$	$2 \cdot r_x^2 \cdot y_{k+1}$
0	-332	1	6	72	768
1	-224	2	6	144	768

2	-44	3	6	216	768
3	208	4	5	288	640
4	-108	5	5	360	640
5	288	6	4	432	512
6	244	7	4	504	384

Vùng 2

Nếu $p_{2,k} \leq 0 \Rightarrow p_{2,k+1} = p_{2,k} + 2.r_y^2.x_{k+1} - 2.r_x^2.y_{k+1} + r_x^2$ và $x_{k+1} = x_k + 1$ và $y_{k+1} = y_k - 1$

Nếu $p_{2,k} > 0 \Rightarrow p_{2,k+1} = p_{2,k} - 2.r_x^2.(y_k - 1) + r_x^2 + r_y^2[(x_{k+1} + \frac{1}{2})^2 - (x_k + \frac{1}{2})^2]$ và $y_{k+1} = y_k - 1$

và $x_{k+1} = x_k$

Điều kiện dừng: $y_{k+1} = 0$

k	$p_{1,k}$	x_{k+1}	y_{k+1}	$2.r_y^2.x_{k+1}$	$2.r_x^2.y_{k+1}$
0	-151	8	2	576	256
1	233	8	1	576	128
2	745	8	0		

6		X	X	X					
5					X	X			
4							X		
3								X	
2									X
1									X
0									X
	0	1	2	3	4	5	6	7	8

Tuần 5: BEZIER (18/11/2021)

Câu 1:Viết pt đường cong Bezier

$$P(u) = \sum_{k=0}^n p_k BEZ_{k,n}(u), \text{ với } 0 \leq u \leq 1$$

(Hint: Tính 4 hàm uốn

$$BEZ_{k,n}(u) = C_n^k \cdot u^k (1-u)^{n-k} = (1-u)BEZ_{k,n-1}(u) + u \cdot BEZ_{k-1,n-1}(u), \text{ với } n > k \geq 1$$

Bezier k=0, n=3

$$BEZ_{0,3}(u) = (1-u)^3$$

k=1, n=3

$$\begin{aligned} BEZ_{1,3}(u) &= (1-u) \cdot BEZ_{1,2}(u) + u \cdot BEZ_{0,2}(u) \\ &= (1-u) \cdot [(1-u) \cdot BEZ_{1,1}(u) + u \cdot BEZ_{0,1}(u)] + u \cdot (1-u)^2 \\ &= (1-u)(1-u) \cdot u^1 + (1-u) \cdot u \cdot (1-u)^1 + u(1-u)^2 = 3u(1-u)^2 \end{aligned}$$

k=2, n=3

$$BEZ_{2,3}(u) = (1-u)BEZ_{2,2}(u) + u \cdot BEZ_{1,2}(u) = (1-u) \cdot u^2 + u[3u(1-u)^2]$$

k=3, n=3

$$BEZ_{3,3}(u) = u^3$$

)

Câu 2: So sánh các điều kiện biên với tính chất của đường cong Bezier

.

Câu 3: Sai phân. Tính $x_k = x(u_k)$

$$/* x_k = x_{k+1} - \Delta x_k = x(u_{k+1}) - (x_{k+1} - x_k) = x(u_k + \delta) - (x(u_{k+1}) -$$

*/

$$\text{denta}^3 x_{k-2} = 6a \cdot \text{theta}^3$$

$$\text{denta}^2 x_{k-1} = \text{denta}^2 x_{k-2} + \text{denta}^3 x_{k-2}$$

$$\text{dentax}_k = \text{dentax}_{k-1} + \text{denta}^2 x_{k-1}$$

$$x_{k+1} = x_k + \text{dentax}_k(1)$$

$$(1) \Rightarrow x_{uk+1} = x_{uk} + \text{dentax}_k$$

$$(1) \Rightarrow x_{uk} + \text{theta} = au^3 k + bu^2 k + cuk + d + \text{dentax}_k$$

$$(1) \Rightarrow a(u_k + \theta)^3 + b(u_k + \theta)^2 + c u_k + 1 + d$$

Tuần 6 BIEZIER (CONT) AND FILLING (25/11/2021)

Câu 1: Vẽ đường cong Biezer theo phương pháp của De Casteljeu. Kiểm chứng method 2 ông có giống nhau(trùng) không (ans: giống)

$$BEZ_{0,2}(t) = (1-t)^2$$

$$\Rightarrow P''(0) = 6(p_2 - 2p_1 - p_0)$$

$$\Rightarrow x(0) = 6(x_2 - 2x_1 - x_0)$$

$$\Rightarrow y(0) = 6(y_2 - 2y_1 - y_0)$$

$$p_0^2(t) = (1-t)p_0^1(t) + tp_1^1(t)$$

$$\Rightarrow p_0^2(t) = (1-t)[(1-t)p_0 + tp_1] + t[(1-t)p_1 + tp_2]$$

$$\Rightarrow p_0^2(t) = (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2$$

Câu 2: Tại sao không tô màu tràn ra được?(nguyên lí Boundary Fill)

Vì lân cận 4

Câu 3: Rã đệ qui Boundary Fill

Áp dụng hàng đợi Queue, BFS

-Bước 1: Cất điểm hạt giống đầu tiên vào kho.

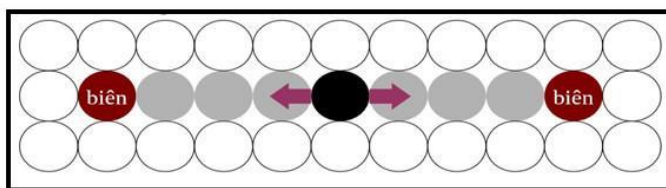
-Bước 2: Lặp nếu kho không rỗng

+Lấy điểm hạt giống.

+Tô điểm hạt giống, sau đó tô loang sang 2 bên.

+Bổ sung những điểm hạt giống mới vào kho từ dòng trên và dòng dưới.

Tiêu chuẩn để làm điểm hạt giống: điểm này chưa được tô màu và không phải điểm biên.



```

#include <conio.h>
#include <winbgim.h>
#include <iostream>
#include <queue>
using namespace std;
struct ToaDo
{
    int x,y;
};
int MauNen;
void NhapDaGiac(int &n,int &x,int &y,ToaDo a[])
{
    cout<<"Nhap so dinh cua da giac n= "; cin>>n;
    for (int i=1;i<=n;i++)
    {
        cout<<"Toa do dinh P["<<i<<"].x= "; cin>>a[i].x;
        cout<<"Toa do dinh P["<<i<<"].y= "; cin>>a[i].y;
    }
    cout<<"Nhap diem (x,y) thuoc da giac:\n";
    cout<<"nhap x="; cin>>x;
    cout<<"nhap y="; cin>>y;
}
void VeDaGiac(int n,ToaDo a[],int color)
{
    setcolor(color);
    for (int i=1;i<=n;i++)
    {
        int j;
        if (i==n) j=1; else j=i+1;
        line(a[i].x,a[i].y,a[j].x,a[j].y);
    }
}
void ToLoang(int x,int y,int color)
{
    // Khai bao queue chua pixel chua duoc to mau
    queue<ToaDo> Q;
    ToaDo m, Tg;
    if (getpixel(x,y)==MauNen && x<getmaxx() && y<getmaxy())
    {
        m.x = x;
        m.y = y;
        putpixel(m.x, m.y, color);
        Q.push(m); // Them 1 diem vao queue, queue size tang 1
        while(Q.empty() == false) //Xet 4 diem xung quanh voi moi diem luu trong queue (neu queue
con phan tu)
        {
            Q.pop();// Xoa 1 diem phia dau queue, queue size giam 1
            //Xet cac diem lan can cua 1 diem
            if(getpixel(m.x+1, m.y) == MauNen)
            {
                putpixel(m.x+1, m.y, color );
            }
        }
    }
}

```

```

        Tg.x = m.x+1;
        Tg.y = m.y;
        Q.push(Tg);// Them 1 diem vao cuoi queue
    }
    if(getpixel(m.x-1, m.y) == MauNen)
    {
        putpixel(m.x-1, m.y, color);
        Tg.x = m.x-1;
        Tg.y = m.y;
        Q.push(Tg);
    }
    if(getpixel(m.x, m.y+1) == MauNen)
    {
        putpixel(m.x, m.y+1, color);
        Tg.x = m.x;
        Tg.y = m.y+1;
        Q.push(Tg);
    }
    if(getpixel(m.x, m.y-1) == MauNen)
    {
        putpixel(m.x, m.y-1, color);
        Tg.x = m.x;
        Tg.y = m.y-1;
        Q.push(Tg);
    }
    m = Q.front();// Dua ve gia tri dau tien cho hang doi
    delay(1);
}
}
}
int main()
{
    int x,y,n,Gd,Gm=VGAMAX;
    ToaDo a[10];
    NhapDaGiac(n,x,y,a);
    Gd=DETECT;
    initgraph(&Gd,&Gm,"");
    VeDaGiac(n,a,15);
    MauNen=getpixel(x,y);
    ToLoang(x,y,10);
    getch();
    closegraph();
}

```

Câu 4: Tiêu chí để tự động xét điểm cực trị (loại bỏ cạnh //Ox) và k cực trị (ngắn cạnh)

Sử dụng tiêu chí sự đổi dấu (đồng biến, nghịch biến) để xét vì tiêu chí thay không phụ thuộc gì vào việc Δx tiến đến 0

$$\Delta y_k = y_k - y_{k-1}$$

$$\Delta y_{k+1} = y_{k+1} - y_k$$

Đồng biến khi: $\Delta y_k * \Delta y_{k+1} > 0$

Nghịch biến khi: $\Delta y_k * \Delta y_{k+1} < 0$

Loại trường hợp đoạn thẳng khi Δy_k hoặc Δy_{k+1} bằng 0.

Không dùng tiêu chí xét: hàm liên tục, tại điểm cực trị thì $f'(x) = 0$ vì như vậy là *lim* rất khó trong việc sử dụng dữ liệu rời rạc

Câu 5: Tính giao điểm dòng quét vs các cạnh (hint: công thức qui nạp tính giao điểm dòng quét với 1 cạnh)

$$Ta\ có: m = \frac{\Delta y}{\Delta x} = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}$$

Mà các dòng quét tăng đều theo y ($y+=1$) $\Rightarrow y_{k+1} - y_k = 1$

$$\Rightarrow x_{k+1} - x_k = \frac{1}{m}$$

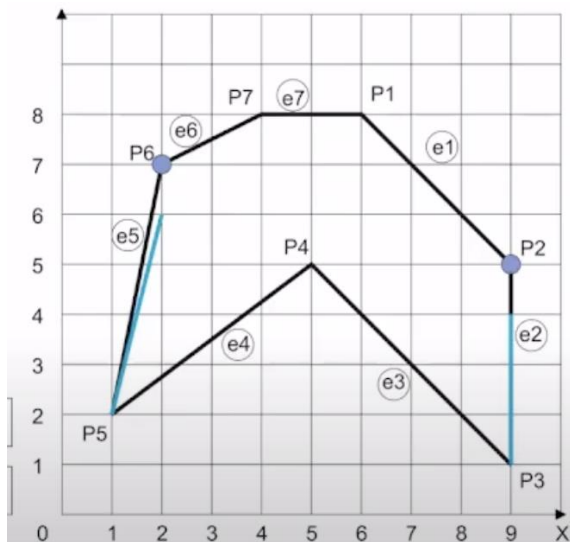
$$\Rightarrow x_{k+1} = x_k + \frac{1}{m}$$

Câu 6: Tại sao phải sort giao điểm theo thứ tự tăng dần trong Scan-line Polygon Fill method

Bởi vì trong quá trình hoạt động (thêm, xóa, sửa), thì các cạnh chỉ bởi Beglist bị xáo trộn thứ tự của x_intersection, đảm bảo tiền đề bất cứ lúc nào cũng tô được từ trái sang phải theo x tăng dần.

Tuần 7 FILLING (CONT) AND CLIPPING : 02/12/2021

Câu 1



Bảng ET[y]

y=1: push e2, e3

4	9	0	->	5	9	-1	
---	---	---	----	---	---	----	--

e2

e3

y=2: push e4, e5

5	1	4/3	->	6	1	1/5	
---	---	-----	----	---	---	-----	--

e4

e5

y=5: push e1

8	9	-1	
---	---	----	--

e1

y=7: push e6

<u>8</u>	<u>2</u>	<u>2</u>	
----------	----------	----------	--

e6

y=8 (Không push vì // Ox)

Scanline y=1

Add e2, e3 to Beglist

4	9	0	->	5	9	-1	
e2			e3				

Sort by x_intersect

4	9	0	->	5	9	-1	
e2			e3				

Fill

$$(9,1) \rightarrow (9,1)$$

Remove y_upper == scanline y

4	9	0	->	5	9	-1	
e2			e3				

Update x_intersect $+= \frac{1}{m}$

4	9	0	->	5	9	-1	
e2			e3				

Scanline y=2

Add e4, e5 to Beglist

4	9	0	->	5	8	-1	->	5	1	$\frac{4}{3}$	->	6	1	$\frac{1}{5}$	->	NIL
e2			e3			e4			e5							

Sort by x_intersect

5	1	$\frac{4}{3}$	->	6	1	$\frac{1}{5}$	->	5	8	-1	->	4	9	0	->	NIL
e4			e5			e3			e2							

Fill

$$x_1 = 1; y_1 = 2$$

$$x_2 = 1; y_2 = 2$$

$$x_3 = 8; y_3 = 2$$

$$x_4 = 9; y_4 = 2$$

Remove $y_{upper} == \text{scanline } y$

5	1	$\frac{4}{3}$	->	6	1	$\frac{1}{5}$	->	5	8	-1	->	4	9	0	->	NIL
e4				e5				e3				e2				

Update $x_{intersect} += \frac{1}{m}$

5	$\frac{7}{3}$	$\frac{4}{3}$	->	6	$\frac{6}{5}$	$\frac{1}{5}$	->	5	7	-1	->	4	9	0	->	NIL
e5				e4				e3				e2				

Scanline $y=3$

Add

5	$\frac{7}{3}$	$\frac{4}{3}$	->	6	$\frac{6}{5}$	$\frac{1}{5}$	->	5	7	-1	->	4	9	0	->	NIL
e4				e5				e3				e2				

Sort by $x_{intersect}$

6	$\frac{6}{5}$	$\frac{1}{5}$	->	5	$\frac{7}{3}$	$\frac{4}{3}$	->	5	7	-1	->	4	9	0	->	NIL
e5				e4				e3				e2				

Fill

$$x_1 = \text{round}\left(\frac{6}{5}\right) = 1; y_1 = 3$$

$$x_2 = \text{round}\left(\frac{7}{3}\right) = 2; y_2 = 3$$

$$x_3 = 7; y_3 = 3$$

$$x_4 = 9; y_4 = 3$$

Remove $y_{upper} == \text{scanline } y$

6	$\frac{6}{5}$	$\frac{1}{5}$	->	5	$\frac{7}{3}$	$\frac{4}{3}$	->	5	7	-1	->	4	9	0	->	NIL
e5				e4				e3				e2				

Update $x_{intersect} += \frac{1}{m}$

6	$\frac{7}{5}$	$\frac{1}{5}$	->	5	$\frac{11}{3}$	$\frac{4}{3}$	->	5	6	-1	->	4	9	0	->	NIL
e5				e4				e3				e2				

Scanline y=4

Add

6	$\frac{7}{5}$	$\frac{1}{5}$	->	5	$\frac{11}{3}$	$\frac{4}{3}$	->	5	6	-1	->	4	9	0	->	NIL
e5				e4				e3				e2				

Sort by $x_intersect$

6	$\frac{7}{5}$	$\frac{1}{5}$	->	5	$\frac{11}{3}$	$\frac{4}{3}$	->	5	6	-1	->	4	9	0	->	NIL
e5				e4				e3				e2				

Fill

$$x_1 = \text{round}\left(\frac{7}{5}\right) = 1; y_1 = 4$$

$$x_2 = \text{round}\left(\frac{11}{3}\right) = 4; y_2 = 4$$

$$x_3 = 6; y_3 = 4$$

$$x_4 = 9; y_4 = 4$$

Remove $y_upper == \text{scanline } y$ e2

6	$\frac{7}{5}$	$\frac{1}{5}$	->	5	$\frac{11}{3}$	$\frac{4}{3}$	->	5	6	-1	->	NIL
e5				e4				e3				

Update $x_intersect += \frac{1}{m}$

6	$\frac{8}{5}$	$\frac{1}{5}$	->	5	5	$\frac{4}{3}$	->	5	5	-1	->	NIL
e5				e4				e3				

Scanline y=5

Add e1 to Beglist

6	$\frac{8}{5}$	$\frac{1}{5}$	->	5	5	$\frac{4}{3}$	->	5	5	-1	->	8	9	-1	->	NIL
e5				e4				e3				e1				

Sort by $x_intersect += \frac{1}{m}$

6	$\frac{8}{5}$	$\frac{1}{5}$	->	5	5	$\frac{4}{3}$	->	5	5	-1	->	8	9	-1	->	NIL
---	---------------	---------------	----	---	---	---------------	----	---	---	----	----	---	---	----	----	-----

e5

e4

e3

e1

Fill

$$x_1 = \text{round}\left(\frac{8}{5}\right) = 2; y_1 = 5$$

$$x_2 = 5; y_2 = 5$$

$$x_3 = 5; y_3 = 5$$

$$x_4 = 9; y_4 = 5$$

Remove $y_upper == \text{scanline } y \text{ e3, e4}$

6	$\frac{8}{5}$	$\frac{1}{5}$	->	8	9	-1	->	NIL
---	---------------	---------------	----	---	---	----	----	-----

e5

e1

Update $x_intersect += \frac{1}{m}$

6	$\frac{9}{5}$	$\frac{1}{5}$	->	8	8	-1	->	NIL
---	---------------	---------------	----	---	---	----	----	-----

e5

e1

Scanline y=6

Add

6	$\frac{9}{5}$	$\frac{1}{5}$	->	8	8	-1	->	NIL
---	---------------	---------------	----	---	---	----	----	-----

e5

e1

Sort by $x_intersect$

6	$\frac{9}{5}$	$\frac{1}{5}$	->	8	8	-1	->	NIL
---	---------------	---------------	----	---	---	----	----	-----

e5

e1

Fill

$$x_1 = \text{round}\left(\frac{9}{5}\right) = 1; y_1 = 6$$

$$x_2 = 8; y_2 = 6$$

Remove $y_upper == \text{scanline } y \text{ e5}$

8	8	-1	->	NIL
---	---	----	----	-----

e1

Update $x_intersect += \frac{1}{m}$

8	7	-1	->	NIL
---	---	----	----	-----

e1

Scanline y=7

Add e6 to Beglist

8	7	-1	->	8	2	2	->	NIL
---	---	----	----	---	---	---	----	-----

e1

e6

Sort by $x_intersect$

8	2	2	->	8	7	-1	->	NIL
---	---	---	----	---	---	----	----	-----

e6

e1

Fill

$$x_1 = 2; y_1 = 7$$

$$x_2 = 7; y_2 = 7$$

Remove $y_upper == scanline\ y$

8	2	2	->	8	7	-1	->	NIL
---	---	---	----	---	---	----	----	-----

e6

e1

Update $x_intersect += \frac{1}{m}$

8	4	2	->	8	6	-1	->	NIL
---	---	---	----	---	---	----	----	-----

e6

e1

Scanline y=8

Add

8	4	2	->	8	6	-1	->	NIL
---	---	---	----	---	---	----	----	-----

e6

e1

Sort by $x_intersect$

8	4	2	->	8	6	-1	->	NIL
---	---	---	----	---	---	----	----	-----

e6

e1

Fill

$$x_1 = 4; y_1 = 8$$

$$x_2 = 6; y_2 = 8$$

Remove $y_upper == scanline\ y$

NIL

Update $x_intersect += \frac{1}{m}$

None

Câu 2: Tìm biểu thức toán thể hiện các trường hợp, mối quan hệ vị trí (nằm ngoài, nằm trong, cắt..) Cohen Sutherland line clipping 8.1.2

outside

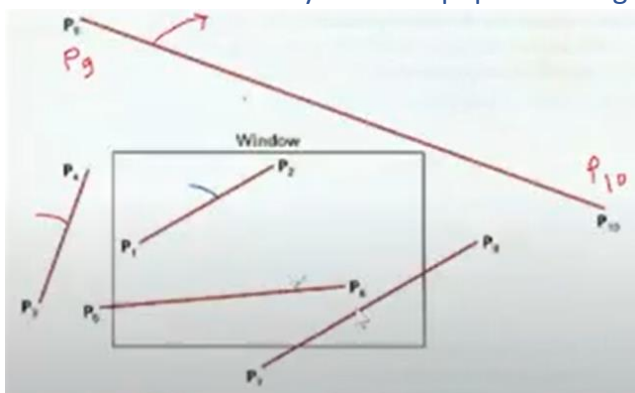
$$\begin{cases} x_1, x_2 < x_{min} \\ x_1, x_2 > x_{max} \\ y_1, y_2 < y_{min} \\ y_1, y_2 > y_{max} \end{cases}$$

inside

$$\begin{cases} x_{min} \leq x_1, x_2 \leq x_{max} \\ y_{min} \leq y_1, y_2 \leq y_{max} \end{cases}$$

intersect (cắt/ clipping) \Leftrightarrow 2 điểm không rơi vào cả 2 trường hợp trong và ngoài

Câu 3: Làm sao cho máy tính biết p9p10 nằm ngoài



chuyển 2 điểm p9,p10 thành 2 mã vùng, mỗi mã vùng gồm 4 bit

mã vùng p9

bit trên	bit dưới	bit trái	bit phải
----------	----------	----------	----------

mã vùng p10

bit trên	bit dưới	bit trái	bit phải
----------	----------	----------	----------

out \Rightarrow (1000 && 1000) || (0100 && 0100) || (0010 && 0010) || (0001 && 0001)

$\Rightarrow ((\text{code 1} \&\& \text{code 2}) \neq 0)$

Câu 4: Hàm encode để biến điểm thành mã vùng

```
encode(p1,wmin,wmax){
code1=0x00
if(p1.x<wmin.x)
code1 = code1 | 0x01;
if(p1.x>wmax.x)
code1=code1 | 0x02;
if(p1.y<wmin.y)
code1=code1 | 0x04;
if(p1.y>wmax.y)
code1=code1 | 0x08;
return code1;
}
```

Câu 5: inside()

```
inside(code1,code2){
if((code1 | code 2 == 0x00))return true
return false
}
```

Câu 6: outside()

```
outside(code1, code2){
if((code1 & code2 !=0))
return true
return false
}
```

Tuần 8 GEOMETRIC TRANSFORMATION (09/12/2021)

Câu 1: Làm sao để tính ra công thức 2D Rotation

$$x' - x_c = (x - x_c)\cos\theta - (y - y_c)\sin\theta$$

$$y' - y_c = (x - x_c)\sin\theta - (y - y_c)\cos\theta$$

Công thức tổng quát:

(T: translation, R: rotation)

$$R(C, \theta) = T_{(\vec{OC})} \cdot R_{(O, \theta)} \cdot T_{(\vec{CO})}$$

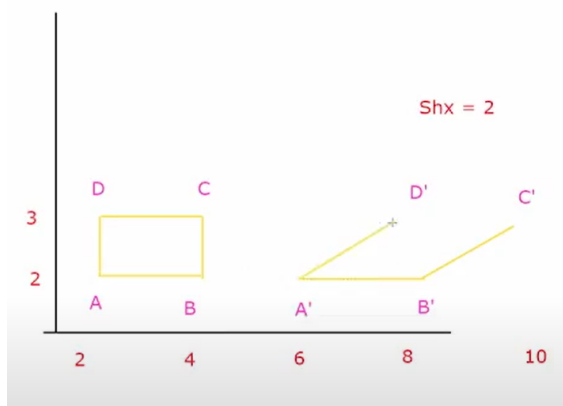
$$\begin{aligned}
&= \begin{bmatrix} 1 & 0 & x_c - x_o \\ 0 & 1 & y_c - y_o \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & x_o - x_c \\ 0 & 1 & y_o - y_c \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta & -\sin\theta & x_c \\ \sin\theta & \cos\theta & y_c \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta & -\sin\theta & -x_c \cdot \cos\theta + y_c \cdot \sin\theta + x_c \\ \sin\theta & \cos\theta & -x_c \cdot \sin\theta - y_c \cdot \cos\theta + y_c \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Câu 2: Ma trận biến đổi co giãn scale tâm C

$$S(C, sx, sy) = T_{\vec{OC}} \cdot S(O, sx, sy) \cdot T_{\vec{CO}}$$

$$\begin{aligned}
&= \begin{bmatrix} 1 & 0 & x_c - x_o \\ 0 & 1 & y_c - y_o \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & x_o - x_c \\ 0 & 1 & y_o - y_c \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} s_x & 0 & x_c \\ 0 & s_y & y_c \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} s_x & 0 & s_x \cdot (-x_c) + x_c \\ 0 & s_y & s_y \cdot (-y_c) + y_c \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Câu 3: Shear Trượt tương đối so với trục Ox



$$x'A = x_A + shx \cdot y_A = 2 + 2 \cdot 2 = 6$$

$$x'B = x_B + shx \cdot y_B = 4 + 2 \cdot 2 = 8$$

$$x'C = x_C + shx \cdot y_C = 4 + 2 \cdot 3 = 10$$

$$x'D = x_D + shx \cdot y_D = 2 + 2 \cdot 3 = 8$$

Câu 4: Quay quanh trục //Ox,Oy,Oz qua tâm C 9.4.2

$$\begin{aligned}
 R(\vec{V}, \theta) &= T^{-1}(P_1\vec{O}) A(\vec{V})^{-1} R(Oz, \theta) A(\vec{V}) T(P_1\vec{O}) \\
 &= \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot R(Ox, \alpha)^{-1} \cdot R(Oy, \beta)^{-1} \cdot R(Oz, \theta) \cdot R(Oy, \beta) \cdot R(Ox, \alpha) \cdot \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{c}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{c}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{c}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & \frac{c}{d} & -\frac{c}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & -\frac{b}{d}y_1 - \frac{c}{d}z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{c}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} d & -\frac{b}{d}a & -\frac{c}{d}a & -x_1d + \frac{b}{d}ay_1 + \frac{c}{d}az_1 \\ 0 & \frac{c}{d} & -\frac{c}{d} & 0 \\ a & b & c & -by_1 - cz_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{c}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} d\cos\theta & -\frac{ab}{d}\cos\theta - \frac{c}{d}\sin\theta & -\frac{ac}{d}\cos\theta + \frac{c}{d}\sin\theta & -x_1d\cos\theta + \frac{ab}{d}y_1\cos\theta + \frac{ac}{d}z_1\cos\theta \\ d\sin\theta & -\frac{ab}{d}\sin\theta + \frac{c}{d}\cos\theta & 0 & 0 \\ a & b & c & -by_1 - cz_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

biết:

$$\sin \alpha = \frac{b}{d}$$

$$\cos \alpha = \frac{\vec{u}' \cdot \vec{u}_z}{|\vec{u}'| \cdot |\vec{u}_z|} = \frac{c}{d}$$

$$d = \sqrt{b^2 + c^2}$$

$$a = \frac{x_2 - x_1}{|V|}, b = \frac{y_2 - y_1}{|V|}, c = \frac{z_2 - z_1}{|V|}$$

$$\vec{V} = \vec{P_1P_2} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

Câu 5: Giải thuật auto phát sinh tập đỉnh nằm trên kinh tuyến vĩ tuyến của quả cầu theo phương trình tọa độ cầu $x^2 + y^2 + z^2 = r^2$ Quadratic Surface

Phương pháp

$$\begin{aligned}
 x &= r \cos \phi \cos \theta, & -\pi/2 \leq \phi \leq \pi/2 \\
 y &= r \cos \phi \sin \theta, & -\pi \leq \theta \leq \pi \\
 z &= r \sin \phi
 \end{aligned}$$

Giải thuật: (hàm có sẵn)

```
glutWireSphere (r, nLongitudes, nLatitudes);
```

Giải thuật:

```

public GgbVector evaluatePoint(double u, double v){

    GgbVector p;

    switch (type){
    case QUADRIC_SPHERE :

        GgbVector n = new GgbVector(new double[] {
            Math.cos(u)*Math.cos(v)*getHalfAxis(0),
            Math.sin(u)*Math.cos(v)*getHalfAxis(0),
            Math.sin(v)*getHalfAxis(0)});

        return (GgbVector) n.add(getMidpoint());

    default:
        return null;
    }

}

public GgbVector evaluateNormal(double u, double v){

    GgbVector n;

    switch (type){
    case QUADRIC_SPHERE :
        return new GgbVector(new double[] {
            Math.cos(u)*Math.cos(v),
            Math.sin(u)*Math.cos(v),
            Math.sin(v)});

    default:
        return null;
    }

}

```

```

public double getMinParameter(int index) {

    switch (type){
    case QUADRIC_SPHERE :
        switch(index){
            case 0: //u
            default:
                return 0;
            case 1: //v
                return -Math.PI/2;
        }

    default:
        return 0;
    }

}

public double getMaxParameter(int index) {

    switch (type){
    case QUADRIC_SPHERE :
        switch(index){
            case 0: //u
            default:
                return 2*Math.PI;
            case 1: //v
                return Math.PI/2;
        }

    default:
        return 0;
    }

}

```

Câu 6: 2 loop kết nối các đỉnh 10.1.2

```

GLUquadricObj *sphere1;
sphere1 = gluNewQuadric ( );
gluQuadricDrawStyle (sphere1, GLU_LINE);

```

Câu 7: Viết cấu trúc dữ liệu để quản lí bảng đỉnh, mặt, cạnh

VERTEX TABLE	EDGE TABLE	SURFACE-FACET TABLE
$V_1:$ x_1, y_1, z_1	$E_1:$ V_1, V_2	$S_1:$ E_1, E_2, E_3
$V_2:$ x_2, y_2, z_2	$E_2:$ V_2, V_3	$S_2:$ E_3, E_4, E_5, E_6
$V_3:$ x_3, y_3, z_3	$E_3:$ V_3, V_1	
$V_4:$ x_4, y_4, z_4	$E_4:$ V_3, V_4	
$V_5:$ x_5, y_5, z_5	$E_5:$ V_4, V_5	
	$E_6:$ V_5, V_1	

```
template <class Polyhedron>
struct Polyhedral_surface_3_type_helper {
    typedef typename Polyhedron::Vertex_handle Vertex_handle;
    typedef typename Polyhedron::Facet_handle Facet_handle;
    typedef typename Polyhedron::Halfedge_handle Halfedge_handle;
    typedef Halfedge_handle Edge_handle;

    typedef typename Polyhedron::Traits::Point_3 Point_3;

    struct Vertex_node {
        typedef std::set<int> Incident_edges; // set of edge id
        Incident_edges incident_edges;
    };

    struct Edge_node {
        typedef std::vector<Edge_handle> Sub_edges;
        typedef std::set<int> Incident_facets; // set of facet id
        Sub_edges sub_edges;
        Incident_facets incident_facets;
    };

    struct Facet_node {
        typedef std::vector<Facet_handle> Sub_facets;
        Sub_facets sub_facets;
    };

    struct Incidence_graph {
        typedef std::vector<Vertex_node> Vertices_nodes;
        typedef std::vector<Edge_node> Edges_nodes;
        typedef std::vector<Facet_node> Facet_nodes;
        typedef typename Vertices_nodes::iterator Vertex_iterator;
        typedef typename Edges_nodes::iterator Edge_iterator;
        typedef typename Facet_nodes::iterator Facet_iterator;
    };
};
```

```

Vertices_nodes vertices;
Edges_nodes edges;
Facet_nodes facets;

int vertex_id(Vertex_iterator vit)
{
    return std::distance(vertices.begin(), vit);
}

int edge_id(Edge_iterator eit)
{
    return std::distance(edges.begin(), eit);
}

int facet_id(Facet_iterator fit)
{
    return std::distance(facets.begin(), fit);
}

```

Tuần 9 PROJECTION TRANSFORM (16/12/2021)

Câu 1: Viết lại d 16:00 chiếu Perspective

$$\begin{aligned}
 Per(C, (R_0, \vec{N})) &= T^{-1}(\vec{CO}).Per(O, (R'_0, \vec{N})).T(\vec{CO}) \\
 &= \begin{bmatrix} 1 & 0 & 0 & c_x \\ 0 & 1 & 0 & c_y \\ 0 & 0 & 1 & c_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 1 & 0 \\ 0 & 0 & d & 0 \\ n_1 & n_2 & n_3 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & c_x \\ 0 & 1 & 0 & c_y \\ 0 & 0 & 1 & c_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} d & 0 & 0 & -dc_x \\ 0 & d & 1 & -dc_y - c_z \\ 0 & 0 & d & -dc_z \\ n_1 & n_2 & n_3 & -(n_1c_x + n_2c_y + n_3c_z) \end{bmatrix} \\
 &= \begin{bmatrix} d + n_1c_x & n_2c_x & n_3c_x & -dc_x - (n_1c_x + n_2c_y + n_3c_z)c_x \\ n_1c_y & d + n_2c_y & 1 + n_3c_y & -dc_y - c_z - (n_1c_x + n_2c_y + n_3c_z)c_y \\ n_1c_z & n_2c_z & d + n_3c_z & -dc_z - (n_1c_x + n_2c_y + n_3c_z)c_z \\ n_1 & n_2 & n_3 & -(n_1c_x + n_2c_y + n_3c_z) \end{bmatrix}
 \end{aligned}$$

Biết:

Tâm chiếu $C(c_x, c_y, c_z)$

Vector pháp tuyến (n_1, n_2, n_3)

$$d = (n_1x_0 + n_2y_0 + n_3z_0) - (n_1c_x + n_2c_y + n_3c_z)$$

Câu 2: Vẽ/mô phỏng quả cầu tâm C có R với mặt phẳng tùy ý

B1: Hàm đặc tả các điểm mặt cạnh

B2: Xây dựng phối cảnh: tự cho tâm chiếu, mặt phẳng chiếu thế nào => có ma trận chiếu tác động vào các tập đỉnh. Tập đỉnh chiếu xong thì nối thành các cạnh

Câu 3: Xác định matrix chiếu gồm những ma trận con nào? với mp chiếu tùy ý, vector chiếu V Parallel (hint: Quay n trùng Oz)

$$Par(\vec{V}, (R_0, \vec{N})) = T(\vec{OR}_0).A^{-1}(\vec{N}).Par(\vec{V}, (O, Oz)).A(\vec{N}).T(\vec{R}_0\vec{O})$$

$$= \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\lambda}{|\vec{N}|} & \frac{n_1n_2}{\lambda|\vec{N}|} & \frac{n_1n_3}{\lambda|\vec{N}|} & 0 \\ 0 & \frac{n_3}{\lambda} & -\frac{n_2}{\lambda} & 0 \\ \frac{n_1}{|\vec{N}|} & \frac{n_2}{|\vec{N}|} & \frac{n_3}{|\vec{N}|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 & -\frac{a}{c} & 0 \\ 0 & 1 & -\frac{b}{c} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\lambda}{|\vec{N}|} & \frac{n_1n_2}{\lambda|\vec{N}|} & \frac{n_1n_3}{\lambda|\vec{N}|} & 0 \\ 0 & \frac{n_3}{\lambda} & -\frac{n_2}{\lambda} & 0 \\ \frac{n_1}{|\vec{N}|} & \frac{n_2}{|\vec{N}|} & \frac{n_3}{|\vec{N}|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \lambda = \sqrt{n_2^2 + n_3^2}$$

Câu 4: Tính depth

$$depth(x, y) = 0 \quad refresh(x, y) = I_{backgnd}$$

$$If \ z > depth(x, y) \ Then \ \{depth(x, y) = z, \ refresh(x, y) = I_{surf}(x, y)\}$$

$$z = \frac{Ax + By + D}{C}$$

$$z' = \frac{A(x+1) + By + D}{C}$$

$$z' = z - \frac{A}{C}$$

Câu 5: Scanline method trong khử mặt khuất: giữ cái nào, bỏ cái nào?

→ flag on, off cho biết cái nào giữ, cái nào bỏ.

Vấn đề tranh chấp: Với 2 flag đều trạng thái on, thì sẽ lấy depth nào lớn hơn.

Tuần 10 (23/12/2021)

Framework của các nhóm

Nhóm Depression:

OpenGL Initialize → OpenGL → OpenGL Draw

OpenGL Initialize gồm:

- `gl.ClearColor(0,0,0,0),`
- `gl.MatrixMode(OpenGL.GL_PROJECTION),`
- `gl.Loadidentity()`

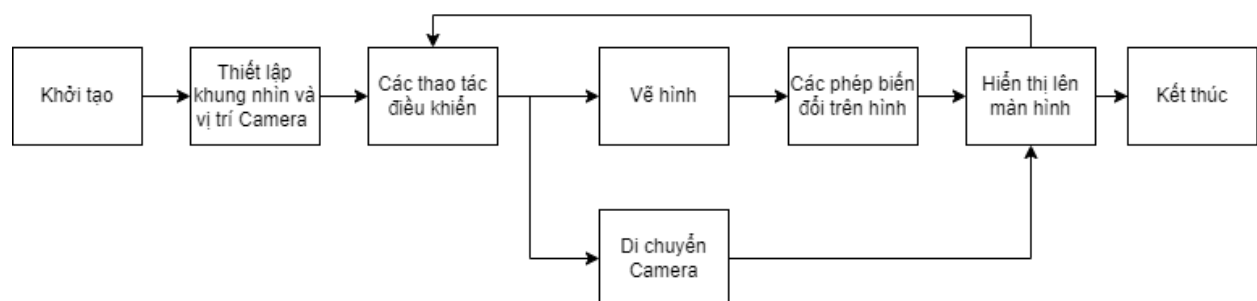
OpenGL Resized gồm:

- `gl.MatrixMode(OpenGL.GL_PROJECTION),`
- `gl.Loadidentity(),`
- `gl.Viewport(x,y,w,h),`
- `gl.Otho2D(l,r,b,t)`

OpenGL Draw gồm:

- `gl.Color(R,G,B),`
- `gl.Clear(),`
- Shape gồm các bước:
 - `gl.PointSize(float size),`
 - `gl.Begin(OpenGL.Mode)`
 - Thuật toán vẽ hình
 - `gl.Vertex(point.X, point.Y)`
 - `gl.End()`
 - `gl.Flush()`

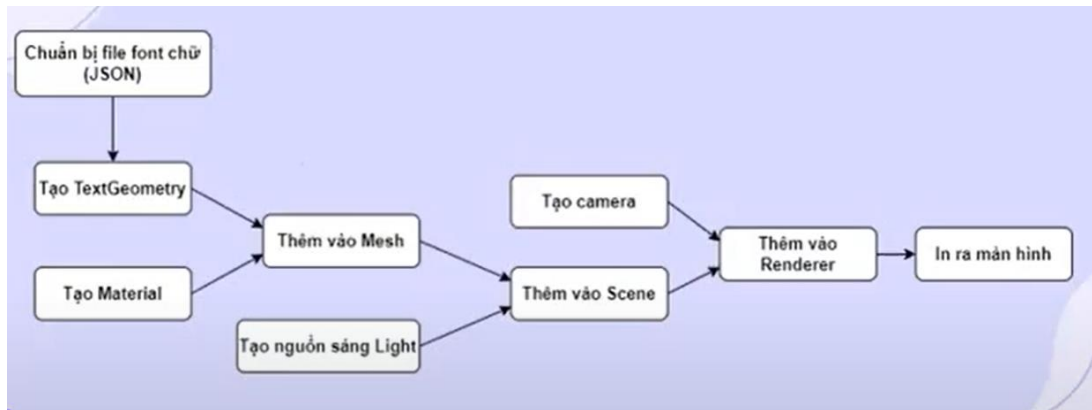
Nhóm TrioCate:



Nhóm 404-noname:

- Get WebGL context
- Create and Compile Shader
- Create Program and attach shader to program
- Setup Attribute pointer for params location in data buffer
- Create buffer (each buffer for each model)
- Draw

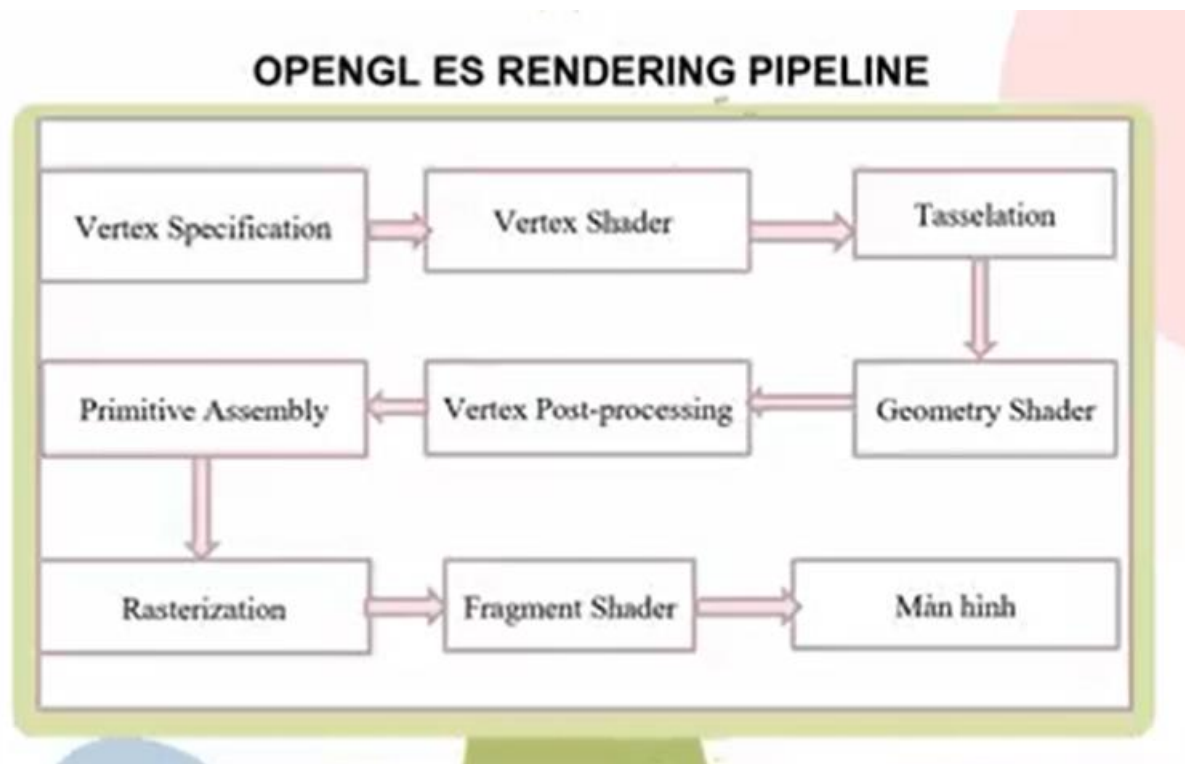
Nhóm TVT:



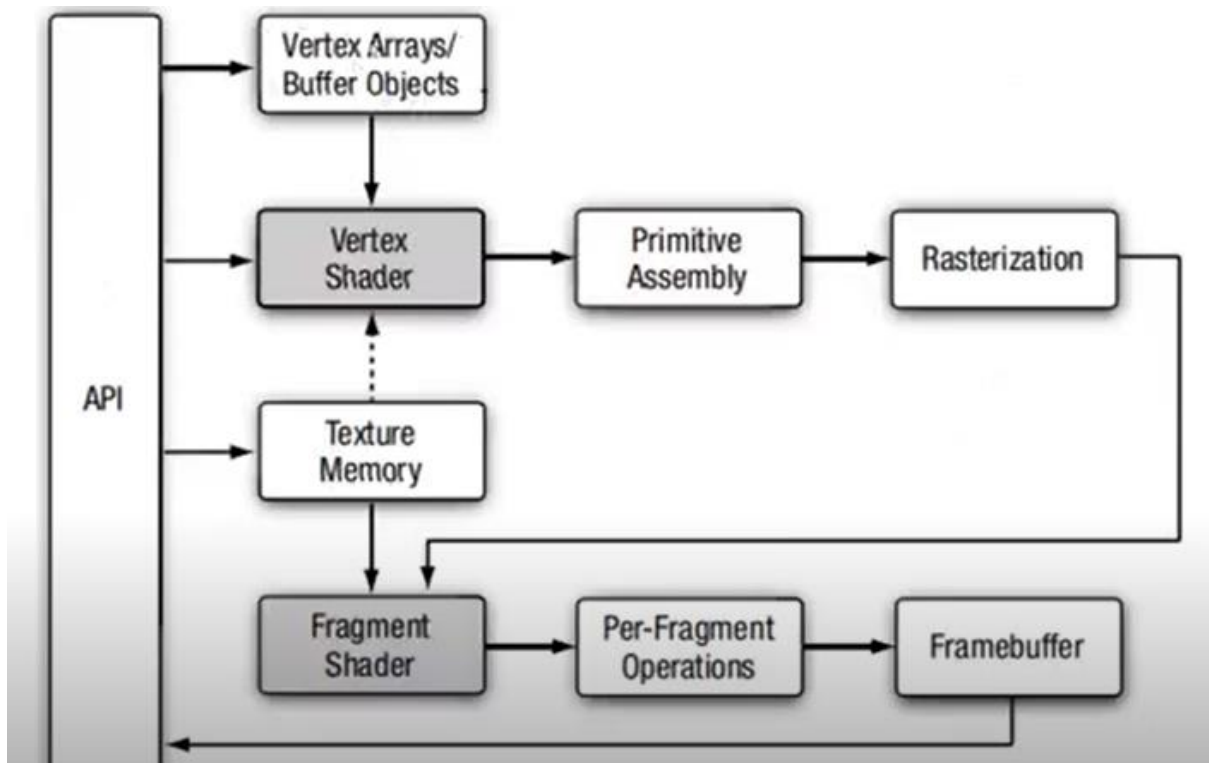
Nhóm VPN:

- B1: Tạo canvas và vẽ bằng context (Khởi động môi trường đồ họa)
- B2: Khởi tạo Viewport
- B3: Tạo các buffer để render (Chuẩn bị dữ liệu trong buffer)
- B4: Tạo một hoặc nhiều ma trận để xác định phép biến đổi từ vùng đệm đỉnh sang không gian màn hình (đẩy ra memory card, buffer màn hình)
- B5: Tạo một hoặc nhiều shader để thực hiện thuật toán vẽ.
- B6: Khởi tạo shader với các tham số.
- B7: Vẽ

Nhóm Noodles:



Nhóm 2 Doses Vaccine:



Nhóm 1:

Point Cloud Data → Oriented Tangent Planes → Signed Distance Function → (modified) Marching Cubes → Polygon Mesh → Visualizer overlay cloud and mesh

Nhóm 3:

Bước 1: Input: Nhập tệp tin dữ liệu point cloud (.txt) dưới dạng ma trận $m \times 3$ với hàng là các tọa độ và cột là x,y,z của tọa độ

Bước 2: biến đổi các dữ liệu point cloud thành các tệp dữ liệu vector pháp tuyến (.pcd)

Bước 3: sử dụng marching cubes algo để tạo tệp dữ liệu lưới (.vtk) từ các dữ liệu vector pháp tuyến

Bước 4: chuyển tệp dữ liệu lưới thành tệp tin .py để mở trên MeshLab