



# LẬP TRÌNH ĐỒ HỌA GAME

Giảng viên:

Email:

SĐT/Zalo:

# Thông tin môn học

- *Mã môn học: CDI1301*
- *Tên môn học: Lập trình đồ họa Game*
- *Tên tiếng Anh: Game Graphics Programming*
- *Số tín chỉ: 03*
- *Lý thuyết: 28 tiết*
- *Bài tập: 08 tiết*
- *Thực hành: 8 tiết*
- *Tự học: 1 tiết*

# Các công việc trong thiết kế và phát triển game

- Theo thông kê của trang HitMaker, hiện có hơn 300 công việc khác nhau có thể làm trong ngành công nghiệp Game
- Bao gồm nhiều mảng khác nhau: Programing, Art, Game Design, Quality Assurance, ...

<https://hitmarker.net/career-advice/the-complete-list-of-gaming-jobs>

## GAMING PROFESSIONS 2025

GAME PROGRAMMING	GAME DESIGN	GAME PRODUCTION
AI • Build & Release Developer Relations • Direction Engine • Gameplay • Graphics Multiplayer • Network • Physics Rendering • Tools • UI	Accessibility • AI • Combat Content • Direction • Economy Encounter • Interaction • Level Mission • Monetization Multiplayer • Narrative • Prototyping Puzzle • Quest • Scripting • Social Systems • Technical • UI World Building • Writing	Accessibility • Archiving Creative Direction • Feature Game Direction Knowledge Management • Live Ops Operations • Production Project Management • Publishing Release Management Studio Management
ART	ANIMATION	AUDIO
2D • 3D • Character • Concept Direction • Environment • Level Lighting • Marketing • Material Model • Outsourcing • Prop Shaders • Storyboarding • Technical Texture • Tools & Pipeline • UI Vehicle • VFX • Weapon	2D • 3D • Cinematic • Gameplay Motion Capture • Particle Programming • Rigging • Scanning Technical • VFX	Composition • Engineering Production • Programming Sound Design • Technical Design Vocal Direction • Voice Design
SOFTWARE ENGINEERING	QUALITY ASSURANCE	LOCALIZATION
Ads • AI • Backend Build & Release • Cloud Data Analytics • DevOps • Direction Engineer Relations • Frontend Full Stack • Manufacturing • Mobile Network • Product Design Project Management • Prompt Reliability • Security • Technical Technical Writing	Accessibility • Analysis Automation • Certification Cinematic • Compatibility Compliance • Designing Engineering • Interruption Performance • Platform • Ratings SDET • Submission • Testing	Editing • Engineering Inclusive Language • Technical Testing • Translation
TALENT	BROADCAST	UX
Agency • Casting • Content Creation Hosting • Interviewing Management • Playing Screen Acting • Streaming Voice Acting	Camera Ops • Floor Management Game Observation • Graphics Lighting • Production • Replay Ops Scriptwriting • Sound Engineering Stream Engineering Studio Management	Accessibility • Design • Engineering Information Architecture • Research Strategy • Usability • Writing
CONTENT & MEDIA	COMPETITIVE	EVENTS
Cinematography • Copy Editing Game Capture • Graphic Design Journalism • Moderation Motion Design • Photography Scriptwriting • Translation Trust & Safety • Video Editing Videography	Analysis • Coaching • Fitness Nutrition • Physiotherapy • Playing Psychology • Refereeing • Scouting Team Management	Audio/Visual • Catering Customer Service Emergency Services Equipment Management Facilities Management Game Observation • Hospitality Logistics • Security
HUMAN RESOURCES	FINANCE & LEGAL	MARKETING
Analysis • Compensation & Benefits Culture • Diversity & Inclusion Employee Relations Employer Branding • Engagement Health & Safety • Mobility People Management • Recruitment Talent Acquisition Training & Development • Wellness	Accounting • Analysis • Auditing Bookkeeping • Compliance • Control Counsel • Fraud • Licensing Mergers & Acquisitions • Payroll Privacy • Reporting Risk Management • Tax	App Store • Brand Campaign Management Community Management Copywriting • Creator • Digital Email • Influencer • Lifecycle Product • Public Relations Social Media • Storefront User Acquisition • Visual
IT	ADMINISTRATIVE	COMMERCIAL
Cyber Security • Network Admin System Admin • Technical Support	Data Entry Executive & Personal Assistance Facilities Support Office Management • Reception	Account Management Advertising Ops Business Development Business Intelligence Communications Consumer Research Customer Support • Distribution Ecommerce • Manufacturing Market Research • Media Buying Merchandising • Monetization Packaging • Partnerships Procurement • Product Management Retail • Sales • Sponsorships Supply Chain • Ticketing
RESEARCH	EDUCATION	Educational Resources
Academic • Consumer • Data Entry Market • Operations	Academic Research Curriculum Design • Lecturing Program Development Program Management Safeguarding • Teaching	HITMARKER IS THE HOME OF GAMING JOBS FIND YOUR DREAM CAREER - HIRE TOP TALENT - MAKE YOUR MARK

# GAMING PROFESSIONS 2025

- Engine Programmer
- Tools Programmer
- Graphics Programmer
- Technical Artist

## GAME PROGRAMMING

- AI • Build & Release
- Developer Relations • Direction
- Engine • Gameplay • Graphics
- Multiplayer • Network • Physics
- Rendering • Tools • UI

## ART

- 2D • 3D • Character • Concept
- Direction • Environment • Level
- Lighting • Marketing • Material
- Model • Outsourcing • Prop
- Shaders • Storyboarding • Technical
- Texture • Tools & Pipeline • UI
- Vehicle • VFX • Weapon

## GAME DESIGN

- Accessibility • AI • Combat
- Content • Direction • Economy
- Encounter • Interaction • Level
- Mission • Monetization
- Multiplayer • Narrative • Prototyping
- Puzzle • Quest • Scripting • Social
- Systems • Technical • UI
- World Building • Writing

## ANIMATION

- 2D • 3D • Cinematic • Gameplay
- Motion Capture • Particle
- Programming • Rigging • Scanning
- Technical • VFX

## GAME PRODUCTION

- Accessibility • Archiving
- Creative Direction • Feature
- Game Direction
- Knowledge Management • Live Ops
- Operations • Production
- Project Management • Publishing
- Release Management
- Studio Management

## AUDIO

- Composition • Engineering
- Production • Programming
- Sound Design • Technical Design
- Vocal Direction • Voice Design

# Mục tiêu môn học

*Khai niệm, thuật ngữ về  
đồ họa máy tính*

*Các kỹ thuật kết xuất (render)  
cảnh trong đồ họa game*

*Chiếu sáng, đổ bóng và vật  
liệu của đối tượng*

*Lập trình Shader, hiệu ứng,  
kết xuất tự nhiên*

*Tối ưu quá trình kết  
xuất cảnh*



# Tài liệu tham khảo

- Allen Sherrod. Game Graphics Programming. Charles River Media, 2008, 673p
- David J.eck, Introduction to Computer Graphics, Hobart and William Smith Colleges, 2021
- Đề cương Lập trình đồ họa Game
- <https://games.ptit.edu.vn>
- Trịnh Thị Vân Anh. Giáo trình kỹ thuật đồ họa. Học viện Công nghệ bưu chính viễn thông, 2018
- <https://catlikecoding.com/unity/tutorials/>
- <https://raytracing.github.io/books/RayTracingInOneWeekend.html>

# Tổng quan chung

## Khái niệm cơ bản

Các thuật ngữ và định nghĩa nền tảng trong đồ họa máy tính.

## 2D & 3D

Phân biệt và hiểu rõ đồ họa 2D và 3D trong phát triển game.

## Toán học nền tảng

Ôn lại và áp dụng các khái niệm toán học quan trọng.

## Hình học cơ bản

Các nguyên lý hình học áp dụng trong đồ họa máy tính.

# Một số khái niệm cơ bản trong đồ họa máy tính

# Đồ họa Máy tính là gì?

- Đồ họa (Graphics) là tập hợp các thiết kế, hình ảnh, chữ viết, ... được tạo ra để trình bày thông tin một cách trực quan
- Đồ họa máy tính là đồ họa được xây dựng trên môi trường số
- Nó kết nối nhiều ngành như đại số, hình học giải tích, hình học họa hình và quang học, đồng thời gắn liền với kỹ thuật máy tính, đặc biệt là việc phát triển phần cứng đồ họa



# Ứng dụng Đồ họa trong Đời sống & Game



**In ấn & Quảng cáo**



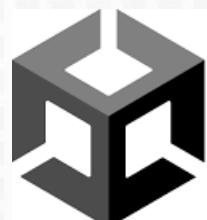
**Phim & Truyền hình**



**Trò chơi điện tử**

# Đồ họa Game là gì?

- Đồ họa game là đồ họa máy tính dành riêng cho game.
- Các trò chơi được xây dựng dựa trên các kỹ thuật kết xuất hình ảnh thường được tích hợp sẵn trong các Game Engine.
- Vai trò:
  - Truyền tải thông tin và tạo ra sự đắm chìm (immersion) cho người chơi
  - Tạo hiệu ứng thị giác tăng cảm giác hấp dẫn cho người chơi



**Unity**



**GODOT**  
Game engine

# Các kỹ thuật kết xuất hình ảnh

- Kết xuất ngoại tuyến (Offline Rendering/Non real-time graphics)
  - Kết xuất thời gian thực (Real-time Rendering)



# So sánh kết xuất ngoại tuyến và kết xuất thời gian thực

## Kết xuất ngoại tuyến

**Mục tiêu:** Chất lượng tối đa.

**Thời gian:** Mỗi khung hình có thể mất vài phút đến vài giờ để kết xuất (render), không yêu cầu thời gian thực.

**Ứng dụng:** Phim hoạt hình (Pixar, DreamWorks), hiệu ứng hình ảnh (VFX) trong phim điện ảnh.

**Ví dụ:** Shrek, Toy Story, Finding Nemo

**Kết luận:** Kết xuất ngoại tuyến luôn vượt trội kết xuất thời gian thực về độ chính xác và chi tiết do không bị ràng buộc bởi thời gian xử lý mỗi khung hình. Do đó, game thường phải tối ưu hóa và đôi khi hy sinh độ chính xác vật lý để đảm bảo tốc độ.

## Kết xuất thời gian thực

**Mục tiêu:** Tốc độ và phản hồi tức thì.

**Thời gian:** Phải xử lý 30-60 khung hình/giây (FPS), tức là chỉ có 16.6 - 33.3 mili giây để vẽ toàn bộ cảnh vật.

**Ứng dụng:** Game, thực tế ảo (VR), thực tế tăng cường (AR).

**Thách thức:** Cân bằng giữa chất lượng hình ảnh và hiệu suất hệ thống.

# ĐẶC THÙ CỦA ĐỒ HỌA TRONG GAME



## Hoạt động thời gian thực

Đồ họa phải được xử lý và hiển thị ngay lập tức để người chơi có thể tương tác liên tục.



## Hướng đến chất lượng đồ họa cao nhất

Tài nguyên đồ họa trong game luôn hướng đến chất lượng cao nhất để có thể thể hiện trọn vẹn nội dung, câu chuyện trong game



## Luôn phải cân bằng

Không có game nào có thể đạt được chất lượng, hiệu năng và tính khả thi tối đa cùng một lúc. "Trade-off" là từ khóa cốt lõi trong lập trình đồ họa game.

Tư duy này giúp các nhà phát triển đưa ra quyết định sáng suốt để tạo ra trải nghiệm game tốt nhất trong giới hạn tài nguyên và thời gian.

# Biểu diễn số và Hệ cơ số

## 3.1. Hệ Nhị phân (Base-2)

- Máy tính sử dụng hệ nhị phân làm ngôn ngữ cơ bản.
- Mỗi bit đại diện cho một lũy thừa của 2, tạo nên các giá trị số.
- Ví dụ: 8 bit có thể biểu diễn  $2^8 = 256$  giá trị khác nhau (từ 0 đến 255).

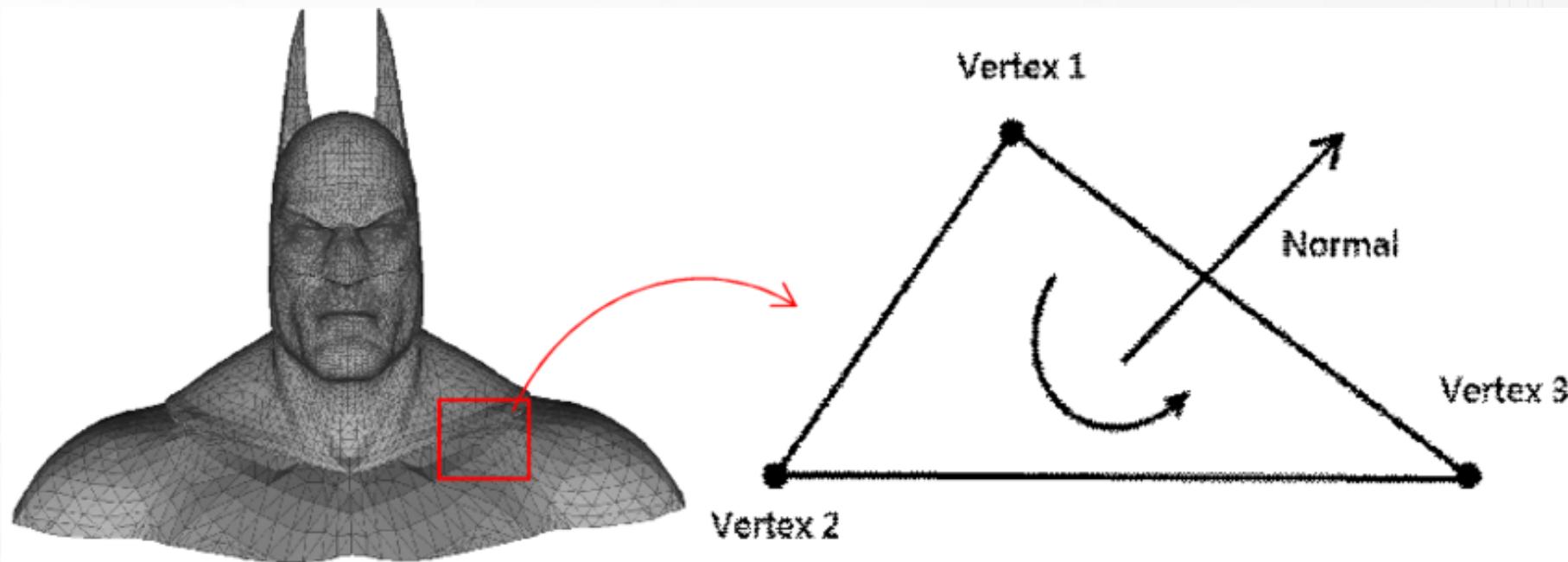
## 3.2. Hệ Thập lục phân (Hexadecimal)

Hệ thập lục phân được dùng để biểu diễn các giá trị phức tạp một cách gọn gàng hơn:

- Biểu diễn màu sắc:** Ví dụ, màu trắng là 0xFFFFFFFF, đen là 0x000000.
- Địa chỉ bộ nhớ:** Giúp lập trình viên dễ dàng làm việc với các vùng nhớ.

Hexadecimal dễ đọc hơn nhị phân, đặc biệt khi làm việc với các giá trị lớn.  
Ví dụ: 0xFF trong hệ thập lục phân tương đương với 255 trong hệ thập phân.

## Model 3D



### ? Vì sao tam giác được dùng nhiều nhất trong đồ họa game?

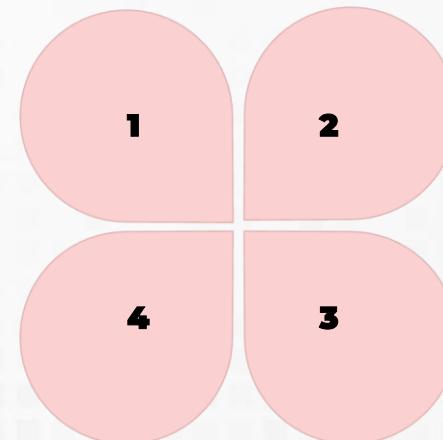
- Luôn đồng phẳng: Ba điểm luôn tạo thành một mặt phẳng duy nhất, giúp tính toán và render dễ dàng.
- Dễ nội suy: Đơn giản để tính toán các giá trị màu, ánh sáng, và texture trên bề mặt.
- GPU xử lý nhanh: Các GPU hiện đại được tối ưu hóa để xử lý hàng tỷ tam giác mỗi giây, đảm bảo hiệu suất cao.
- Tam giác hóa: Mọi polygon (đa giác) đều có thể được chia nhỏ thành các tam giác (tam giác hóa).

## Vertex – Điểm có Dữ liệu

Trong đồ họa 3D, một Vertex không chỉ là một điểm trong không gian mà còn mang theo nhiều thông tin quan trọng khác.

### Position (Vị trí)

Tọa độ (X, Y, Z) của đỉnh trong không gian 3D.



### Texture Coordinate

Tọa độ (U, V) chỉ ra cách texture sẽ được ánh xạ lên bề mặt vật thể.

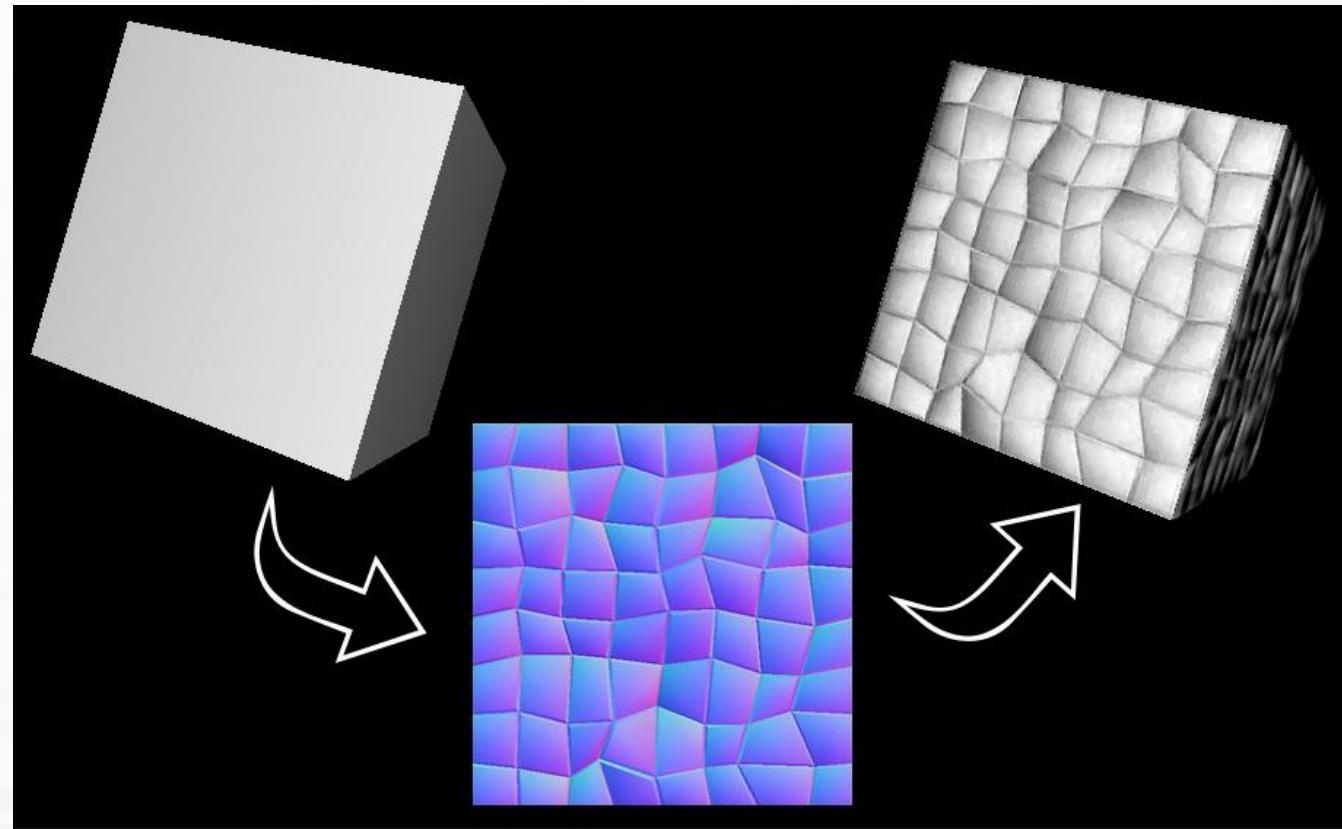
### Normal (Pháp tuyến)

Vector chỉ hướng vuông góc với bề mặt tại đỉnh, quyết định cách ánh sáng tương tác với vật thể.

### Color (Màu sắc)

Màu sắc riêng biệt được gán cho đỉnh

? Nếu thiếu normal, điều gì xảy ra?



#### ❑ ? Nếu thiếu normal, điều gì xảy ra?

Nếu thiếu thông tin normal, vật thể sẽ không phản ứng đúng với ánh sáng, dẫn đến bề mặt trông phẳng lì, không có chiều sâu và không chân thực.

- Vertex là **đầu vào chính của vertex shader**, nơi các thuộc tính của đỉnh được xử lý.
- Mỗi thuộc tính thêm vào cho vertex sẽ làm **tăng bộ nhớ và băng thông cần thiết**.
- Đây là một **trade-off** quan trọng giữa chất lượng hình ảnh và hiệu suất.

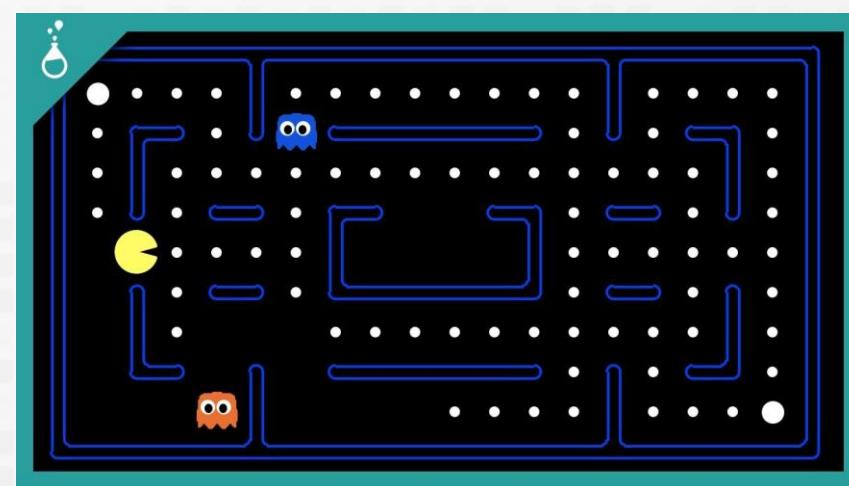
# Đồ họa 2D và Đồ họa 3D

# Đồ Họa 2D

Đồ họa 2D là nền tảng của nhiều tựa game kinh điển và vẫn giữ vững vị thế trong ngành công nghiệp game hiện đại. Nó tập trung vào việc tạo ra hình ảnh trên một mặt phẳng, mang lại phong cách nghệ thuật độc đáo và trải nghiệm chơi game riêng biệt.

## Đặc Điểm Đồ Họa 2D

- Làm việc trong **không gian 2 chiều** (chiều rộng và chiều cao).
- Không có chiều sâu thật sự, các đối tượng được sắp xếp theo lớp để tạo cảm giác gần/xa.
- Các trò chơi 3D cũng không thể tách rời đồ họa 2D



# Khái niệm về Sprite

Sprite là một khái niệm trung tâm trong đồ họa 2D, là các hình ảnh đại diện cho từng phần tử trong game nhân vật, vật thể hoặc môi trường. Các hoạt ảnh của sprite được lưu dưới dạng các Sprite Sheet giúp game 2D trở nên sống động và tăng tính tương tác.



## Đại Diện Nhân Vật

Sprite thường được dùng để hiển thị các nhân vật chính và phụ, với nhiều tư thế và hoạt ảnh khác nhau.



## Vật Thể & Môi Trường

Các vật phẩm, chướng ngại vật, hoặc các yếu tố môi trường tĩnh và động cũng được tạo từ sprite.



## Linh Hoạt Biến Đổi

Sprite có thể được di chuyển, thay đổi kích thước (scale) và xoay (rotate) để tạo ra các hiệu ứng hình ảnh và tương tác đa dạng trong game.

## Đồ Họa 3D

Đồ họa 3D đã cách mạng hóa ngành công nghiệp game, mang đến những trải nghiệm hình ảnh sống động, chân thực và khả năng tương tác sâu sắc mà đồ họa 2D khó lòng sánh được. Từ những tựa game bắn súng góc nhìn thứ nhất đến các thế giới mở rộng lớn, 3D graphics đã định hình lại cách chúng ta chơi và trải nghiệm game.

## Không Gian 3D

- Sử dụng các mô hình toán học trong không gian 3 chiều:** Các vật thể được xây dựng từ lưới đa giác (polygon mesh) trong hệ tọa độ X, Y, Z, cho phép biểu diễn các đối tượng với chiều sâu và khối lượng.
- Cho phép người chơi thay đổi góc nhìn tự do:** Khả năng xoay camera, di chuyển trong môi trường 3D mang lại cảm giác nhập vai và kiểm soát cao hơn.
- Tạo ra sự chân thực nhờ ánh sáng và đổ bóng động:** Hệ thống chiếu sáng và đổ bóng phức tạp mô phỏng thế giới thực, tăng tính thẩm mỹ và độ tin cậy của hình ảnh.



# HIỂN THỊ HÌNH ẢNH 3D TRÊN MÀN HÌNH 2D

- Dựa vào khoảng cách từ camera đến các đối tượng 3D để vẽ lại hình ảnh 3D trên màn hình
- RayTracing, Rasterization



# Nhắc lại các khái niệm toán học

## MỘT SỐ NHÁNH TOÁN HỌC QUAN TRỌNG

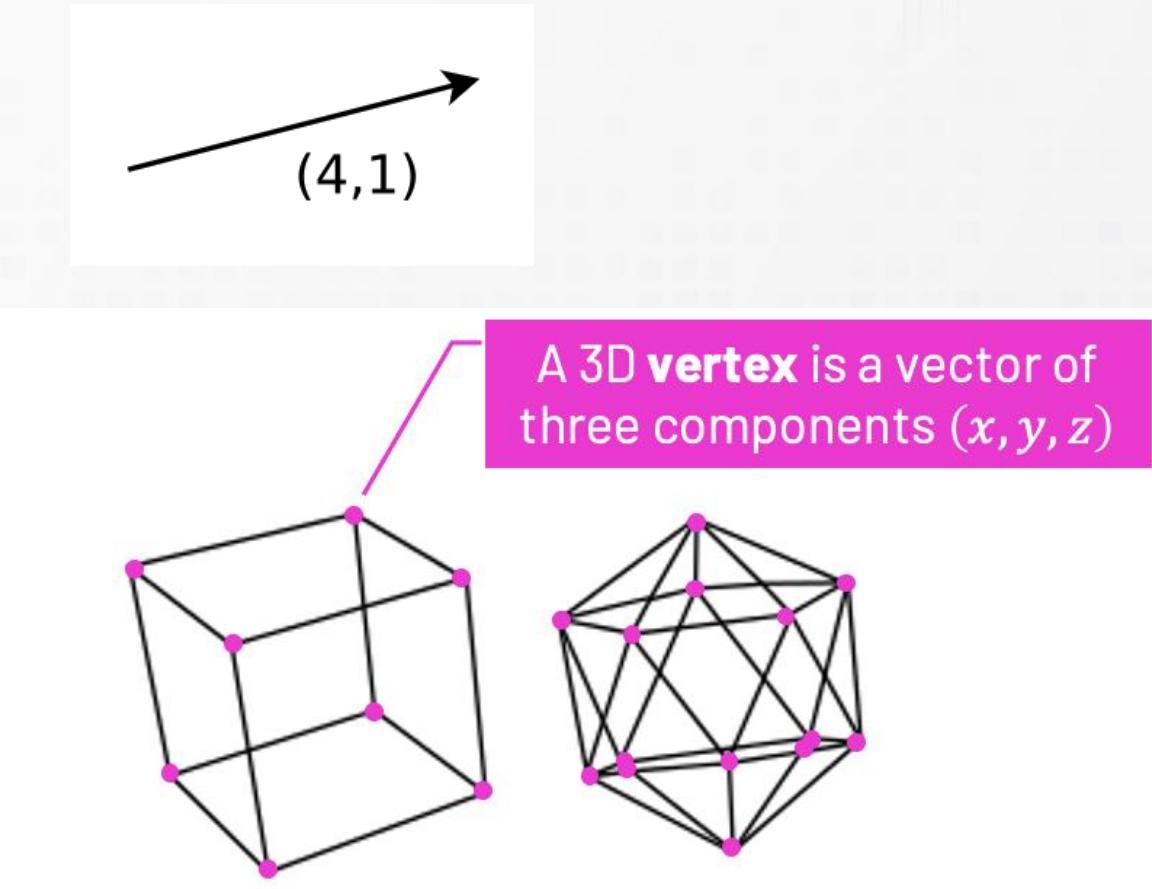
Một số nhánh toán học quan trọng có nhiều ứng dụng

- Đại số tuyến tính (Linear Algebra)
- Hình học (Geometry)
- Toán rời rạc (Discrete Mathematics)
- Xác suất thống kê (Probabilities and Statistics)
- Giải tích (Calculus)

# VECTOR

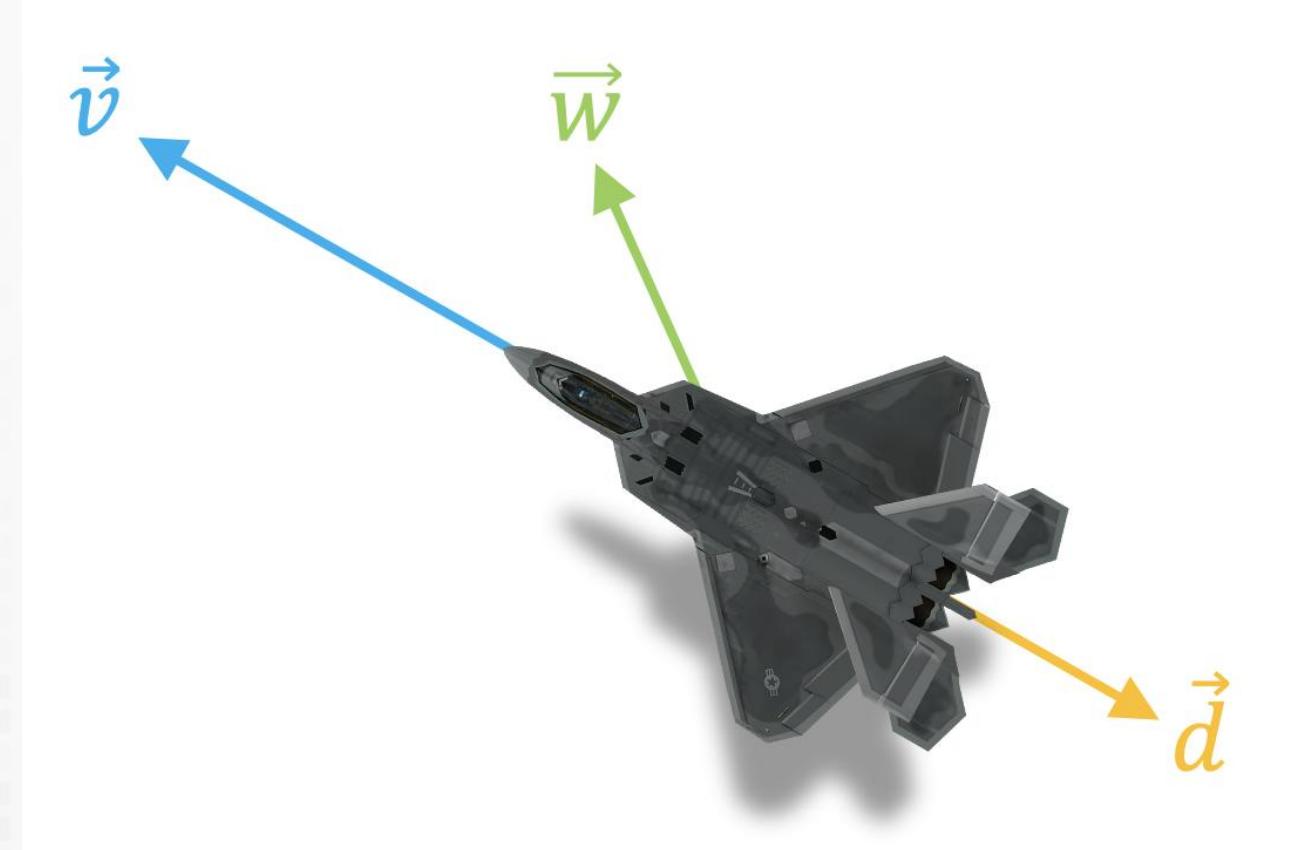
## Vector

- Là đối tượng toán học cơ bản được sử dụng trong hầu hết các game và game engine
- Phần lớn các vector trong lập trình game là 2D ( $x,y$ ), 3D ( $x,y,z$ ), trong một số trường hợp đặc biệt là 4D ( $x,y,z,w$ ).



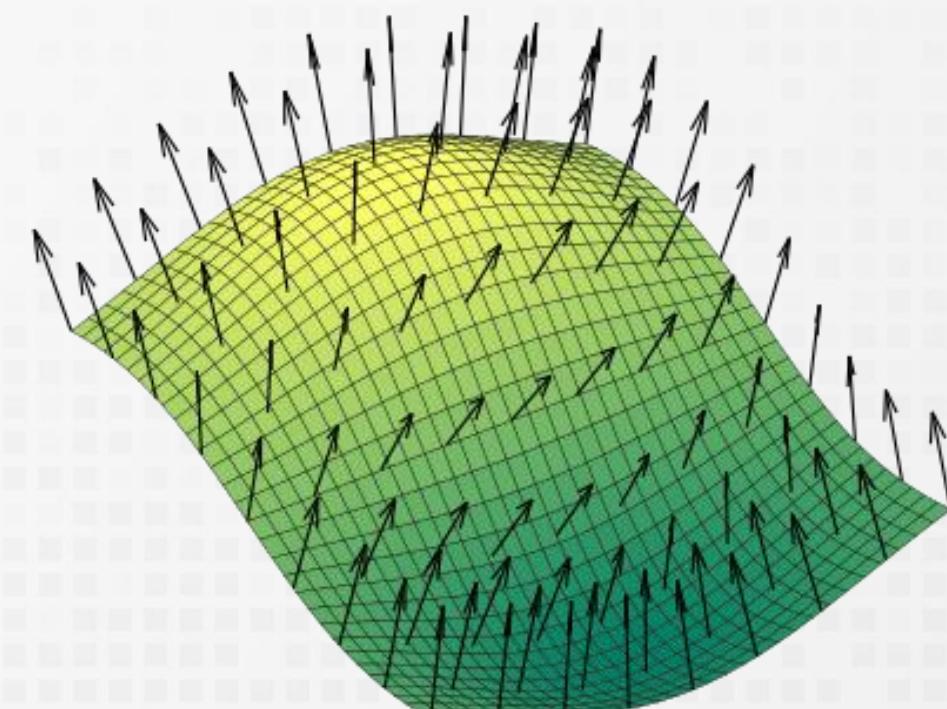
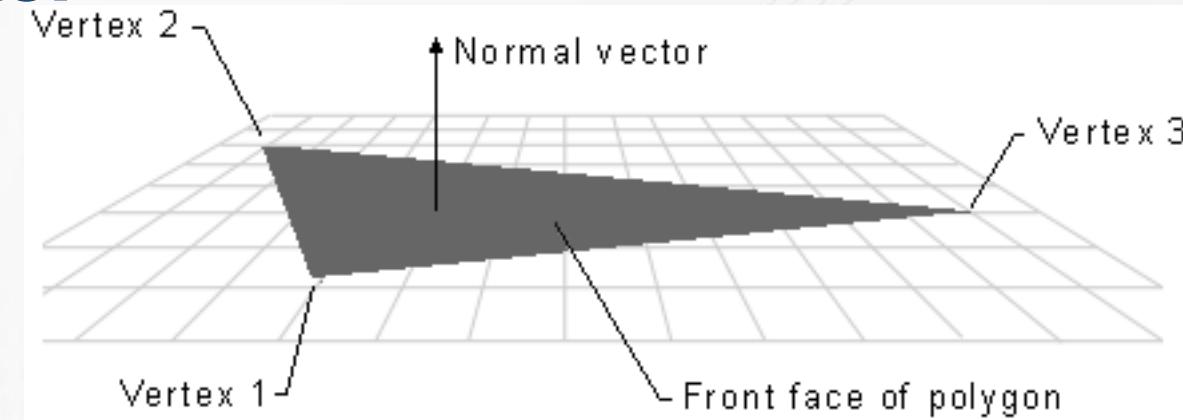
## Vector

- Dùng để biểu diễn một lực, vận tốc, gia tốc, ma sát,... của đối tượng trong thế giới game
- Dễ thực hiện các phép cộng, trừ, nhân vô hướng,...



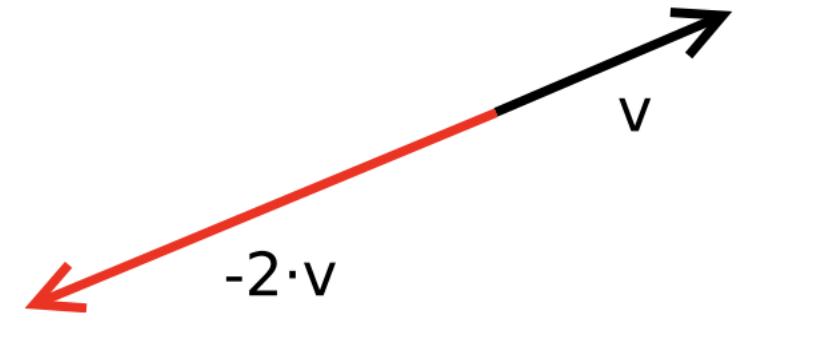
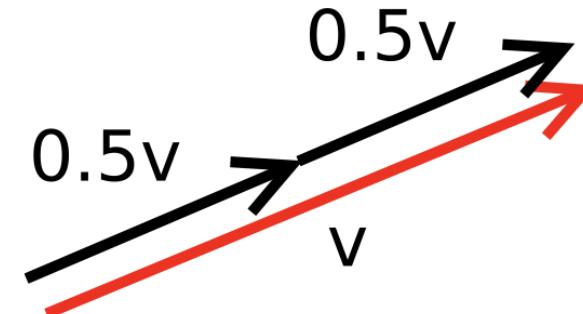
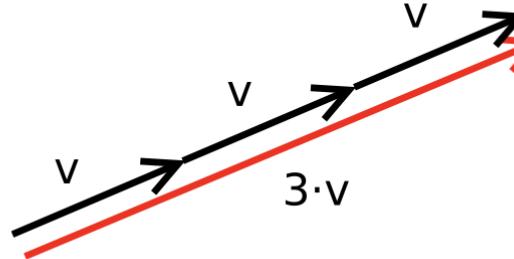
## Vector

- Còn được sử dụng để biểu diễn các bề mặt 3D. Trong đó mỗi phần tử của bề mặt là một đa giác, mỗi đa giác này có một normal vector biểu diễn hướng của bề mặt



# Vector

- Phép nhân
  - Tăng giảm độ lớn vector (scaling)
  - Đảo hướng vector (xoay 180 độ)

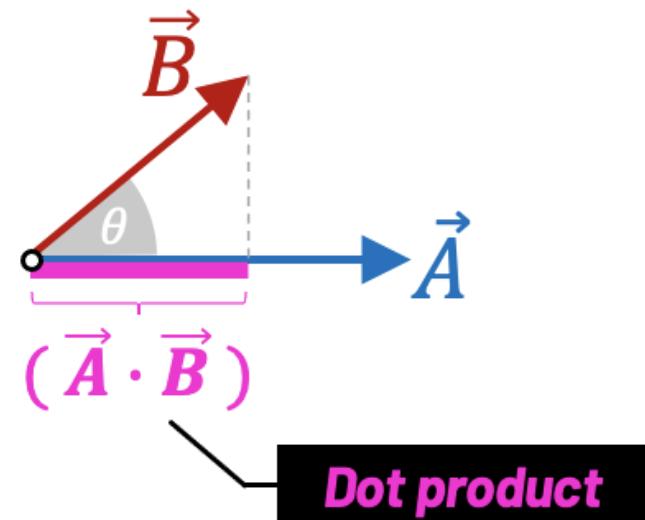


# Tích vô hướng Dot - Product

## Tích vô hướng

- Được sử dụng để tính độ dài của một vector khi chiếu lên một vector khác
- Được sử dụng để so sánh về hướng giữa 2 vector:
  - Cùng hướng nếu tích  $> 0$
  - Vuông góc nếu tích  $= 0$
  - Ngược hướng nếu tích  $< 0$

$$\vec{A} \cdot \vec{B} = |\vec{A}| |\vec{B}| \cos \theta$$

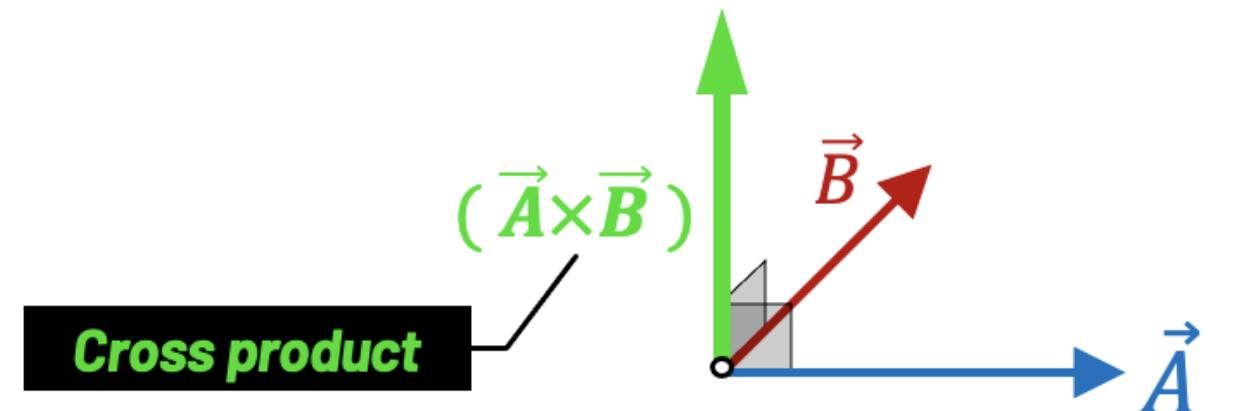


## Tích có hướng (Cross – Product)

Dùng để tính vector pháp tuyến của mặt phẳng được đại diện bởi 2 vector cho trước.

Tập các vector pháp tuyến sử dụng để biểu diễn các bề mặt cong.

$$\vec{A} \times \vec{B} = |\vec{A}| |\vec{B}| \sin \theta$$



Cross product

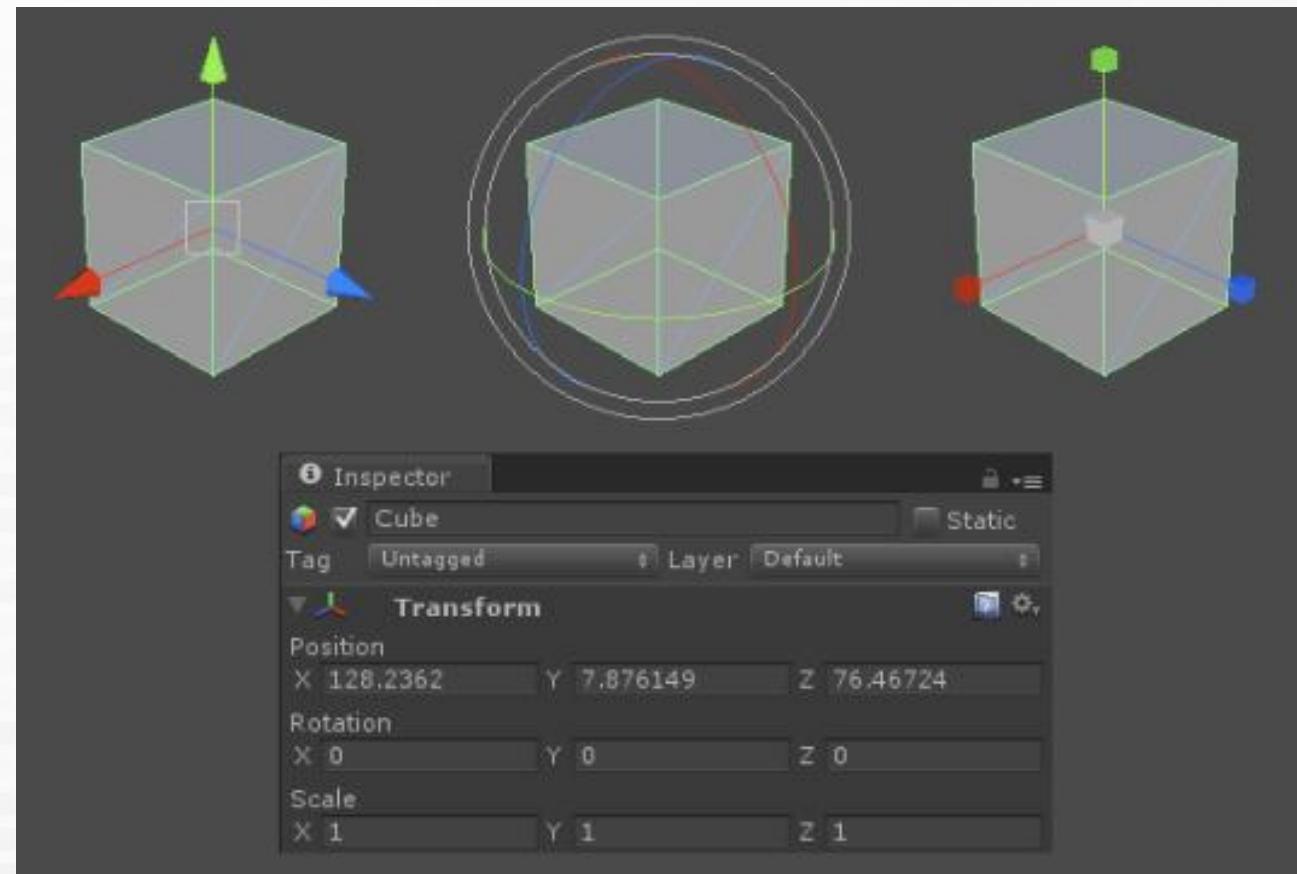
# MA TRẬN

## CÁC ỨNG DỤNG

Là công cụ được sử dụng rất rộng rãi trong game, cụ thể:

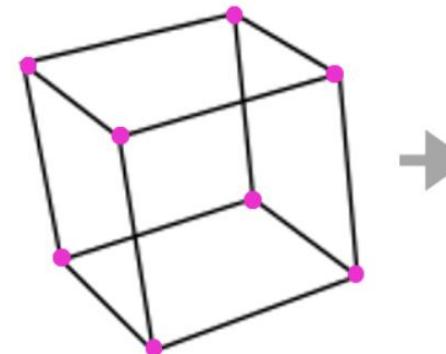
- Phép chiếu, chuyển vị, tỉ lệ cho các đối tượng hình
- Thực hiện phép xoay đối tượng trong không gian
- Chuyển đổi các hệ màu
- Tạo dữ liệu mật mã

# CÁC ỨNG DỤNG

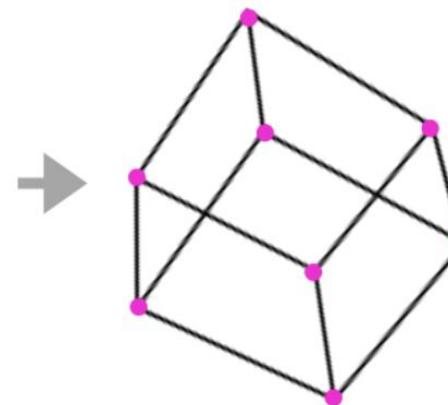


## PHÉP XOAY

- Nhân từng đỉnh của vật thể với ma trận biến đổi (rotate, scale, translate) để tính toán tọa độ mới của đỉnh



```
for (v in vertices) {  
    [cos(theta) -sin(theta) 0]  
    [sin(theta) cos(theta) 0]  
    [0 0 1] * v  
}
```

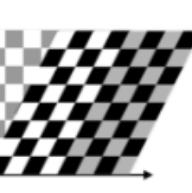
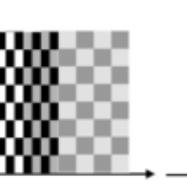
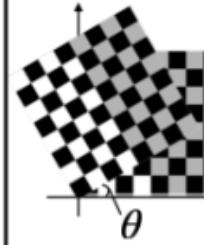


# ỨNG DỤNG PHÉP BIỂN ĐỔI AFFINE

- 3 ứng dụng chính: Xoay (rotation), tỉ lệ (scale), chuyển vị (translation)

Name	Rotate	Translate	Scale	Shear
$M$	$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

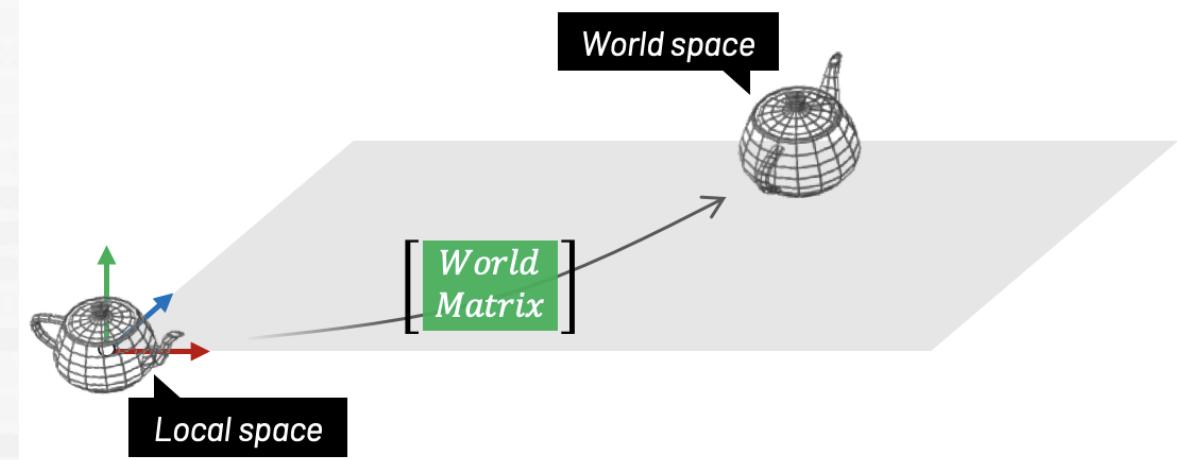
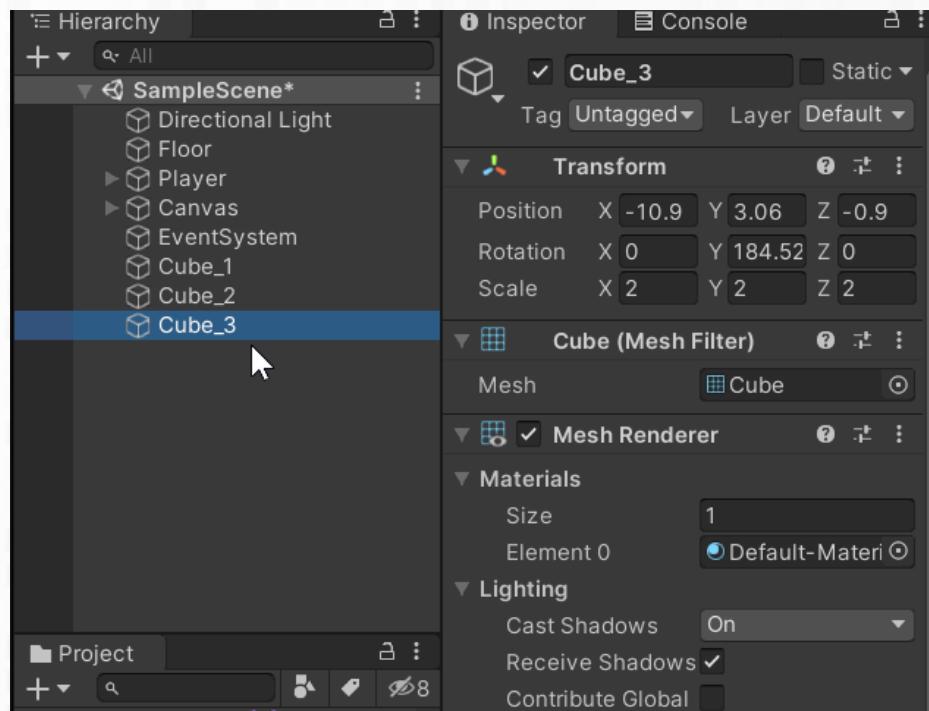
Example



$$\begin{bmatrix} Translation \\ Matrix \end{bmatrix} * \begin{bmatrix} Rotation \\ Matrix \end{bmatrix} * \begin{bmatrix} Scale \\ Matrix \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

# BIỂU DIỄN SỰ CHUYỂN ĐỔI TRỰC TỌA ĐỘ

- Là tập hợp các phép xoay, dịch chuyển và tỉ lệ để chuyển đổi đối tượng giữa 2 không gian tọa độ.
- Được sử dụng bởi các engines



## 4. HÌNH HỌC

# ỨNG DỤNG CỦA HÌNH HỌC

- Được ứng dụng nhiều để xử lý các va chạm giữa các đối tượng trong thế giới game
- Để sinh các màn chơi có yếu tố đồ họa
- ...

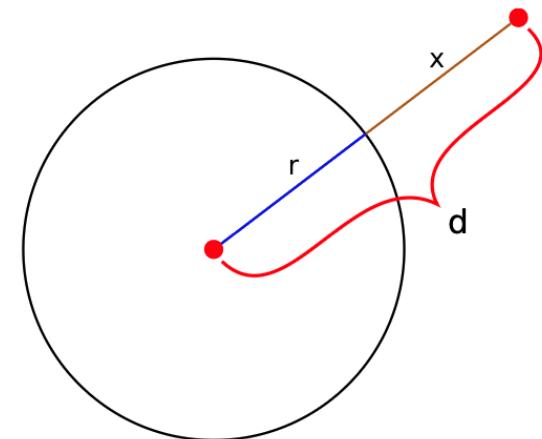


Figure 117: Reference image for Point-Circle Collision detection

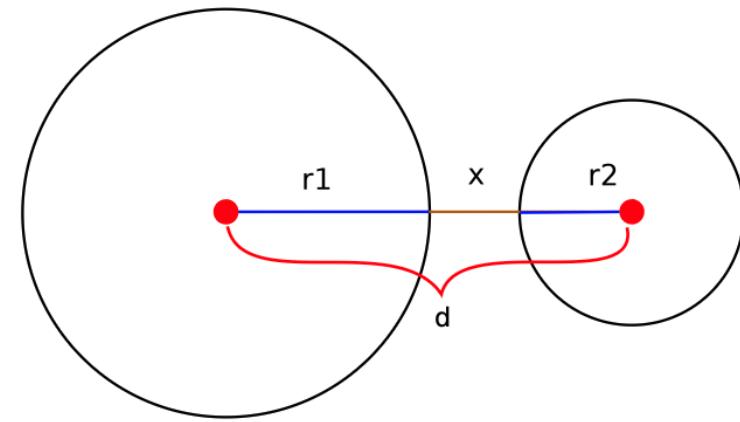


Figure 118: Reference image for Circle-Circle collision detection

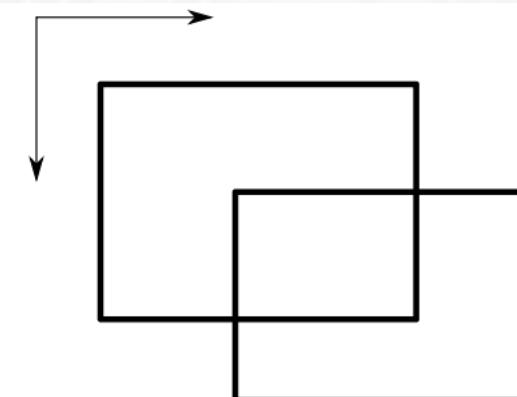


Figure 119: Example used in the AABB collision detection

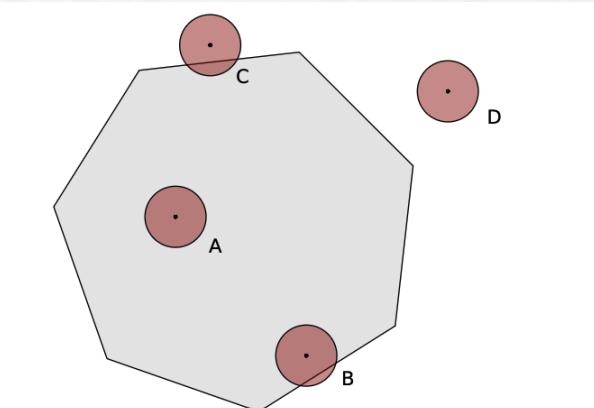


Figure 133: Example image used for circle/polygon collision detection

# ĐƯỜNG THẲNG, ĐA GIÁC LỒI – LÕM

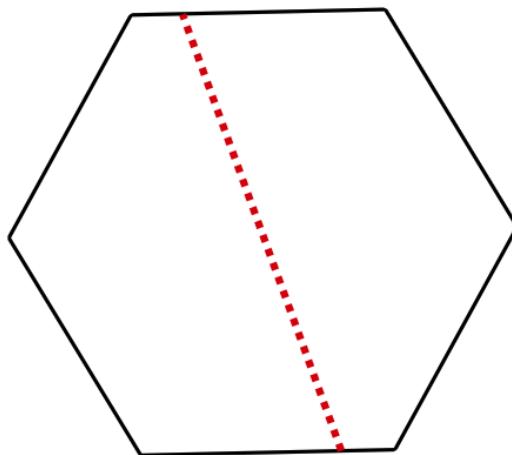


Figure 6: Example of a convex shape

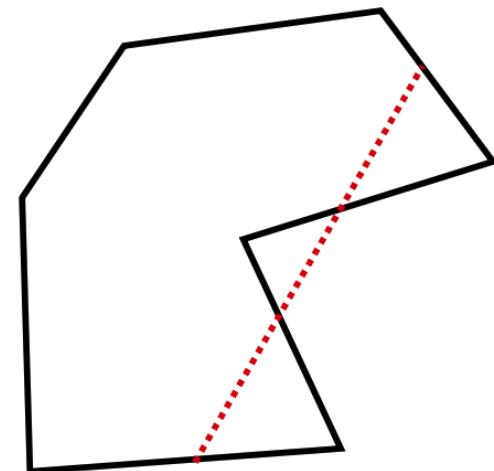


Figure 7: Example of a concave shape

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

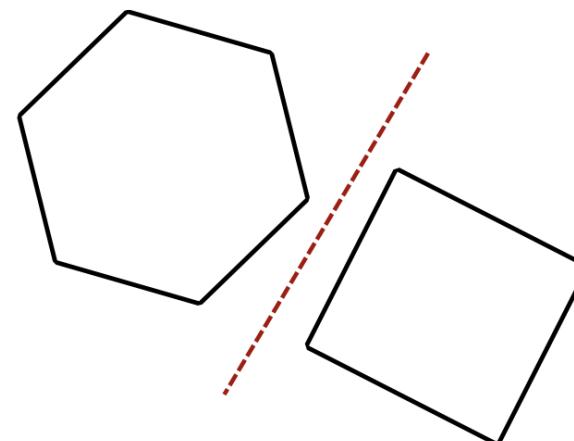
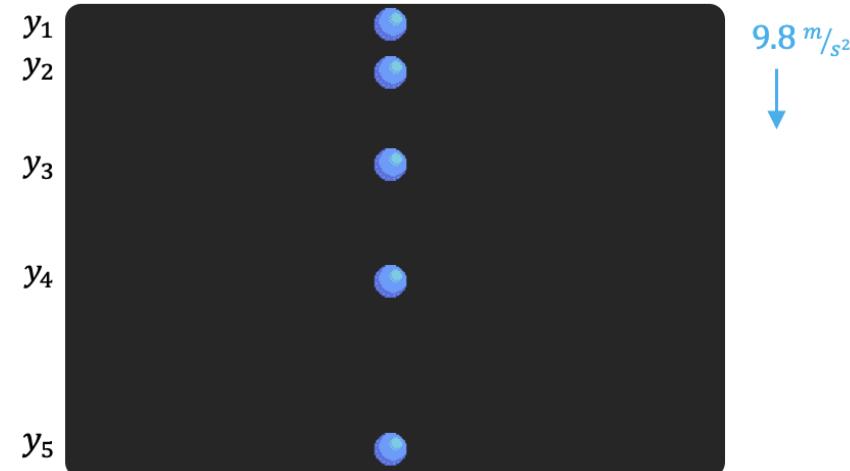
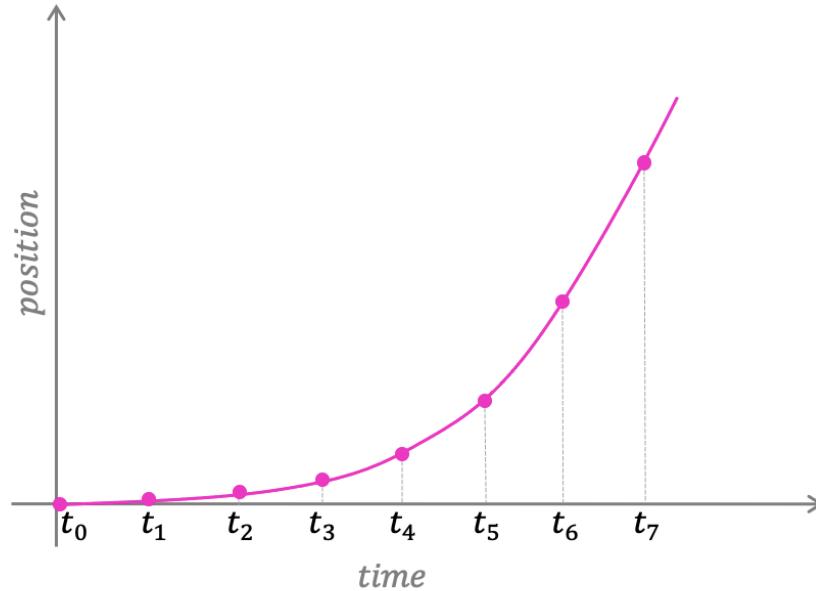


Figure 165: Example of how you can draw a line between two convex non-colliding polygons

- Đa giác lồi (convex) là đa giác có thể vẽ một đường thẳng bất kỳ (trừ cạnh và đường đi qua góc)
- Ngược lại, nếu chỉ cần có 1 đường thẳng cắt đa giác ở nhiều hơn 2 điểm sẽ được gọi là đa giác lõm (non-convex hoặc concave)

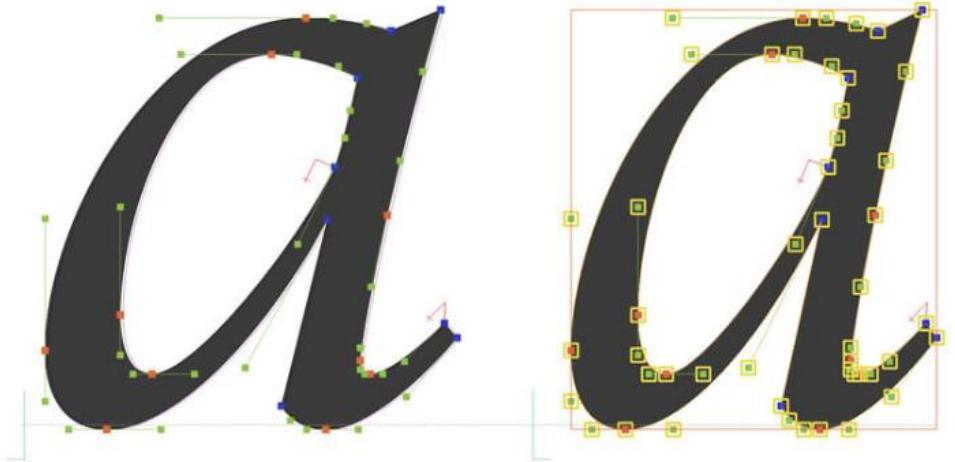
## 5. GIẢI TÍCH

# MÔ PHỎNG VẬT LÝ

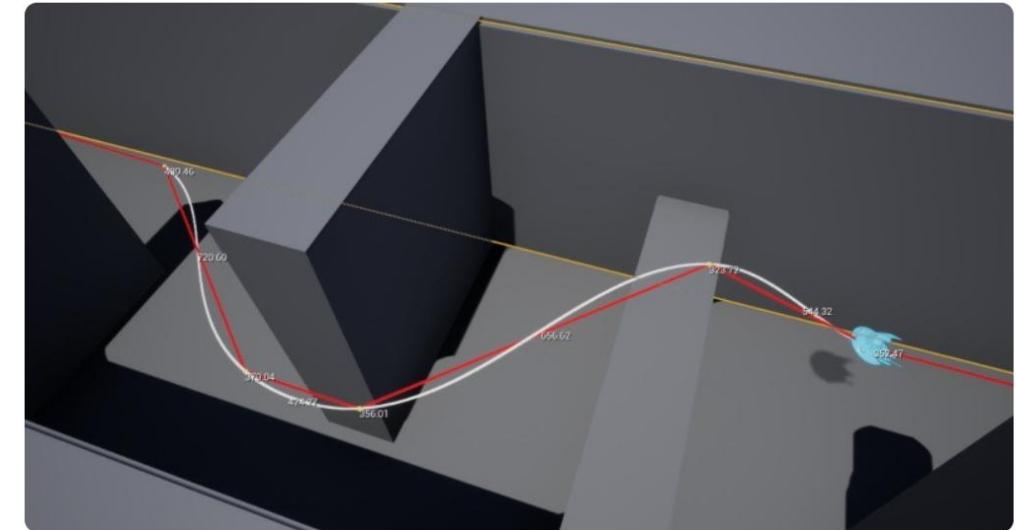


- Giải tích được sử dụng rất nhiều trong mô phỏng các cơ chế vật lý. Ví dụ, tích phân được sử dụng để ước tính vị trí tiếp theo của các đối tượng game ở các khung hình tiếp theo dựa trên gia tốc và vận tốc của chúng.

# VẼ CÁC ĐƯỜNG CONG



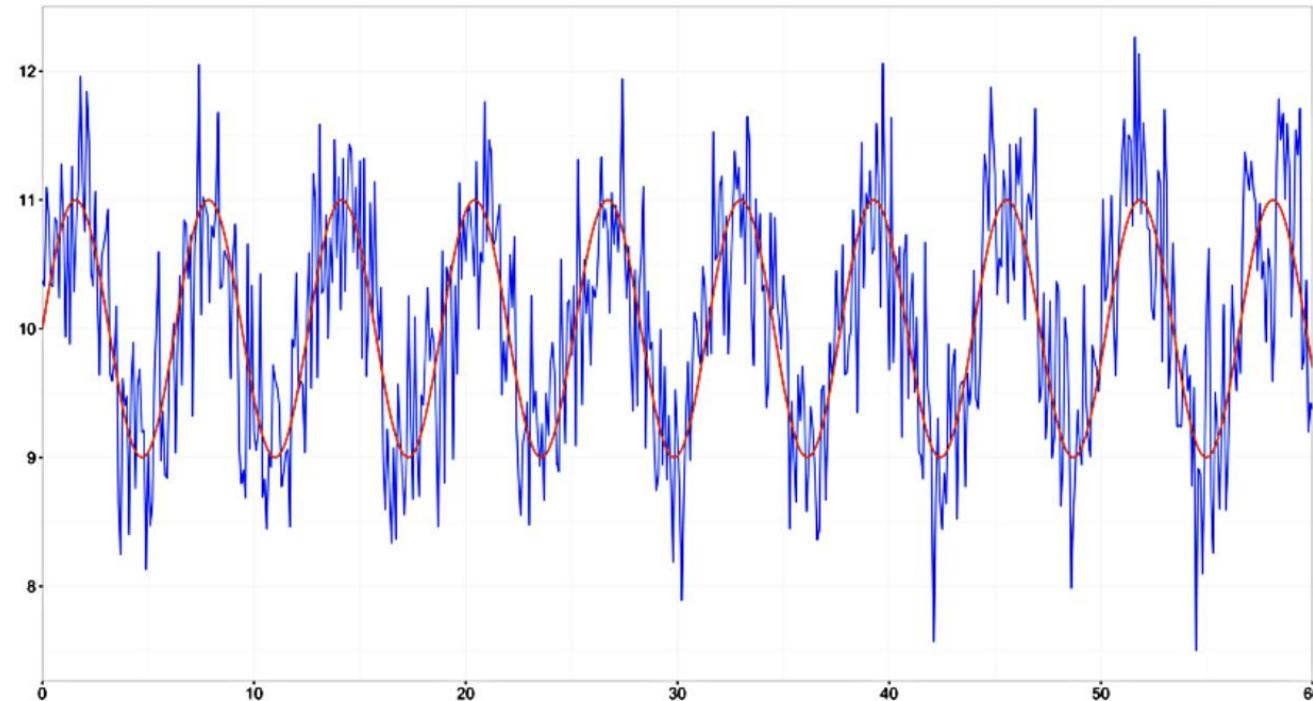
TrueType's curves are quadratic B-splines.



Splines can define a path for an NPC to follow.

- Các hàm Splin (Spline) để vẽ các đường cong dựa trên các đường gấp khúc như Font chữ hoặc đường di chuyển của NPC.
- Các đường cong phổ biến được sử dụng trong phát triển game như Catmull-Rom, Bezier và B-Spline.

# XỬ LÝ ÂM THANH, HÌNH ẢNH



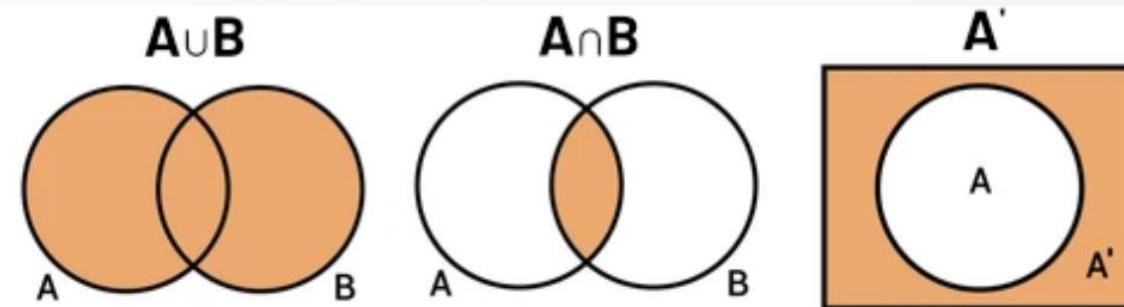
Calculus is used in signal processing applications to filter noise from image and audio.

Bất kỳ phương pháp xử lý tín hiệu số nào đều phải ứng dụng giải tích. Ví dụ điển hình là Biến đổi Fourier (Fourier transforms). Kỹ thuật này được dùng cho rất nhiều ứng dụng trong xử lý tín hiệu trong miền tần số (lọc nhiễu, trộn,...).

## 6. TOÁN RỜI RẠC

## ỨNG DỤNG CỦA TỔ HỢP

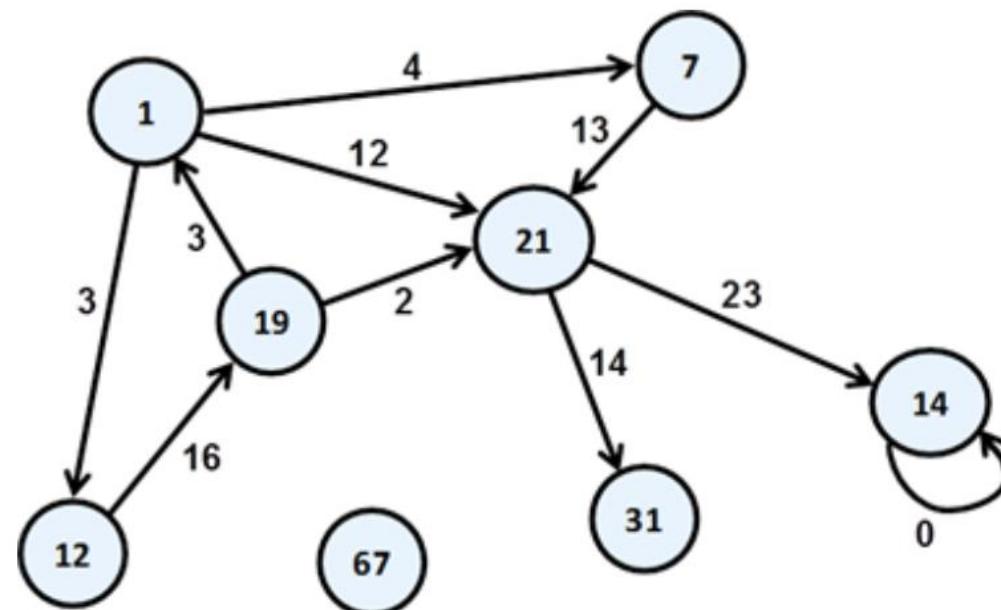
- Các bài toán đề cập đến như là : bài toán đếm, bài toán liệt kê, bài toán tồn tại, nguyên lý Dirichlet, nguyên lý cực hạn. Sau đó còn có tổ hợp, chỉnh hợp, hoán vị, số Sterling, số Catalan,... Các lý thuyết tổ hợp là nền tảng cho lý thuyết tính toán, độ phức tạp, ...



A Venn Diagram representation of the union, intersection, and complement of sets A and B.

# ỨNG DỤNG CỦA LÝ THUYẾT ĐỒ THỊ

- Toán rời rạc sẽ đề cập tới khái niệm đồ thị, những loại đồ thị khác nhau, các thuật toán trên đồ thị ( DFS, BFS, tìm đường đi ngắn nhất, thuật toán tham lam...), những bài toán có thể giải trên đồ thị ( hay mô hình hóa chúng bằng đồ thị để giải quyết), ứng dụng của đồ thị vào các bài toán thực tế ( như là sắp xếp lịch thi, ghép cặp, tìm đường đi ngắn nhất....)



Graph theory has many applications in programming and system design.

## 7. XÁC SUẤT THỐNG KÊ

# ỨNG DỤNG CỦA XÁC SUẤT THỐNG KÊ

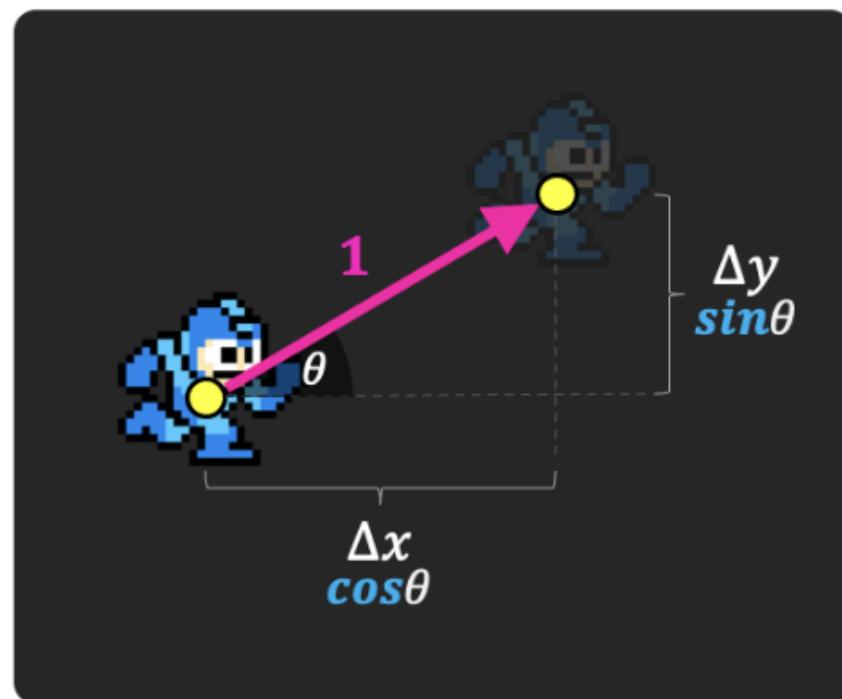
Cải thiện trải nghiệm người dùng:

- **Tính đổi mới (Innovation):** Sử dụng xác suất để tạo ra các trải nghiệm chơi độc đáo và sáng tạo. Nó cũng đồng thời làm cho game có chiều sâu và phức tạp hơn, mang đến trải nghiệm tươi mới và thách thức cho người chơi.
- **Khả năng chơi lại (Replayabilities):** Trong quá trình chơi, người chơi sẽ gặp các tổ hợp mới, giữ cho game cuốn hút trong một quá trình dài từ đó tăng khả năng giữ chân người chơi (player retention) và truyền miệng đến người chơi mới (word-of-mouth recommendations).
- **Chiều sâu chiến thuật (Strategic Depth):** Các cơ chế xác suất thường yêu cầu người chơi tư duy chiến lược và đưa ra các quyết định có tính toán. Họ cần phân tích sự khác biệt, đánh giá rủi ro và tính toán các phần thưởng tiềm năng. Điều này cũng sẽ đem đến sự mới mẻ và thoả mãn khi họ cảm thấy các quyết định có ảnh hưởng trực tiếp đến game

## 8. LƯỢNG GIÁC

# XÁC ĐỊNH VỊ TRÍ CỦA ĐỐI TƯỢNG

- Xác định vị trí tiếp theo của nhân vật dựa trên góc và vận tốc di chuyển:



Finding the of the player movement from sine and cosine.

# BIỂU DIỄN DAO ĐỘNG

- Dùng hình sin để biểu diễn các dòng chảy, các bề mặt chất lỏng (Fluid)

