

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



TRẦN DUY HIẾU

**NGHIÊN CỨU SỬ DỤNG KỸ THUẬT PROCEDURAL CONTENT
GENERATION (PCG) ĐỂ PHÁT TRIỂN TỰ ĐỘNG CÁC MÀN
CHƠI TRONG LẬP TRÌNH GAME**

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

HÀ NỘI – 2025

Đề án tốt nghiệp được hoàn thành tại:
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **PGS. TS. Ngô Quốc Dũng**

Phản biện 1:

Phản biện 2:

Đề án tốt nghiệp sẽ được bảo vệ trước Hội đồng chấm đề án tốt nghiệp thạc sĩ
tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: giờ ngày tháng năm

MỞ ĐẦU

Ngày nay, ngành công nghiệp game đã trở thành một phần không thể thiếu của giải trí toàn cầu. Người chơi ngày càng mong đợi các trò chơi có thể giới rộng lớn, cốt truyện phong phú và trải nghiệm chơi mới mẻ mỗi khi quay lại. Điều này đòi hỏi các nhà phát triển game phải tìm cách tạo ra lượng lớn nội dung mà vẫn đảm bảo chất lượng và tính sáng tạo.

Procedural Content Generation (PCG) là một phương pháp sử dụng thuật toán để tự động tạo ra các nội dung số mà không cần sự can thiệp thủ công. Trong ngành công nghiệp game, PCG đóng vai trò quan trọng trong việc tạo ra các yếu tố như địa hình, màn chơi, cốt truyện, nhân vật, và thậm chí là âm thanh[2]. Khác với phương pháp truyền thống, PCG giúp tạo ra nội dung phong phú, đa dạng và có tính ngẫu nhiên cao, từ đó nâng cao trải nghiệm của người chơi.[3]

Procedural Content Generation (PCG) đáp ứng được nhu cầu tạo ra lượng lớn nội dung bằng cách cho phép tự động tạo ra nội dung game, từ cảnh quan, địa hình đến các nhiệm vụ và thử thách trong game. Việc sử dụng PCG không chỉ giúp tiết kiệm thời gian và công sức mà còn mở ra khả năng tạo ra những trải nghiệm chơi độc đáo, mang tính ngẫu nhiên và bất ngờ, giúp trò chơi có khả năng lôi kéo người chơi quay lại. Với PCG, các màn chơi có thể được tạo ra tự động dựa trên các thuật toán và quy tắc nhất định, giúp giảm thiểu công việc lặp đi lặp lại và cho phép đội ngũ phát triển tập trung vào những khía cạnh sáng tạo hơn của trò chơi. Ngoài ra, việc sử dụng PCG còn giúp giảm thiểu lỗi trong quá trình thiết kế màn chơi. Các thuật toán PCG có thể được lập trình để tuân thủ các quy tắc nhất định, đảm bảo rằng các màn chơi được tạo ra đều có cấu trúc hợp lý, độ khó phù hợp và không gây ra những lỗi nghiêm trọng ảnh hưởng đến trải nghiệm của người chơi.

Mục tiêu của đề tài này là nghiên cứu các thuật toán PCG và phát triển hệ thống ứng dụng kỹ thuật PCG để sinh ra các màn chơi tự động, đảm bảo sự đa dạng, thú vị và thách thức cho người chơi mà không cần sự can thiệp thủ công nhiều từ phía nhà phát triển

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 Xu thế phát triển của ngành công nghiệp trò chơi điện tử và nhu cầu tự động hóa thiết kế màn chơi

1.1.1. Xu thế phát triển của ngành công nghiệp game hiện đại

Ngành công nghiệp trò chơi điện tử trong thế kỷ 21 đang chứng kiến sự tăng trưởng mạnh mẽ cả về quy mô thị trường, số lượng người dùng và công nghệ phát triển. Trong bối cảnh đó, các nhà phát triển game buộc phải thay đổi chiến lược, từ việc tập trung vào đồ họa và cốt truyện, chuyển sang chú trọng đến trải nghiệm người dùng, sự cá nhân hóa và khả năng cập nhật nội dung nhanh chóng.

Một trong những xu thế nổi bật hiện nay là sự phát triển của các trò chơi có tính cá nhân hóa cao, khả năng tương tác sâu và môi trường chơi được cập nhật liên tục. Thay vì chỉ tập trung vào đồ họa hoặc cốt truyện, các nhà phát triển hiện nay chú trọng vào thiết kế trải nghiệm người dùng thông qua hệ thống gameplay phong phú, cơ chế chơi linh hoạt và đặc biệt là sự đa dạng trong hệ thống màn chơi (level design).. Để đáp ứng được nhu cầu đó, việc phát triển và duy trì kho nội dung màn chơi chất lượng cao đang trở thành một thách thức không nhỏ đối với các studio game. Việc này đặt ra những yêu cầu cao hơn đối với khâu thiết kế và phát triển nội dung game.

1.1.2. Các thách thức trong việc phát triển màn chơi trong trò chơi điện tử hiện nay

Trong bối cảnh ngành công nghiệp trò chơi điện tử ngày càng phát triển mạnh mẽ, thiết kế màn chơi (level design) đang đóng vai trò then chốt trong việc nâng cao chất lượng trải nghiệm của người dùng.

Tuy nhiên, quá trình phát triển màn chơi truyền thống hiện đang đối mặt với nhiều thách thức cả về kỹ thuật lẫn chi phí, đặc biệt trong các dự án game quy mô lớn hoặc có tính liên tục cao. Những khó khăn này đặt ra nhu cầu cấp thiết về các giải pháp mới nhằm tối ưu hóa quy trình thiết kế nội dung, trong đó kỹ thuật Procedural Content Generation (PCG) nổi lên như một hướng đi đầy triển vọng.

Thứ nhất, chi phí và thời gian phát triển là một rào cản lớn trong thiết kế màn chơi. Đối với các trò chơi thương mại (AAA), việc xây dựng một màn chơi có thể tiêu tốn hàng tuần đến hàng tháng, yêu cầu sự phối hợp chặt chẽ giữa các bộ phận thiết kế, lập trình và đồ họa để đảm bảo đồng bộ về mặt logic và thẩm mỹ.

Thứ hai, vấn đề về sự đa dạng và chất lượng nội dung cũng là một yếu tố đáng lưu ý. Trong các trò chơi có số lượng lớn màn chơi như game di động hoặc game platform truyền thống, việc thiết kế nội dung thủ công thường dẫn đến tình trạng lặp lại về bố cục, thử thách và trải nghiệm.

Thứ ba, yêu cầu cập nhật nội dung liên tục trong mô hình kinh doanh “Game as a Service” (GaaS) cũng đặt ra áp lực lớn lên việc đảm bảo tiến độ, chất lượng và tính nhất quán về mặt logic của các màn chơi mới trong khi vẫn duy trì tốc độ phát triển cao là một thách thức lớn đối với mô hình thiết kế thủ công truyền thống

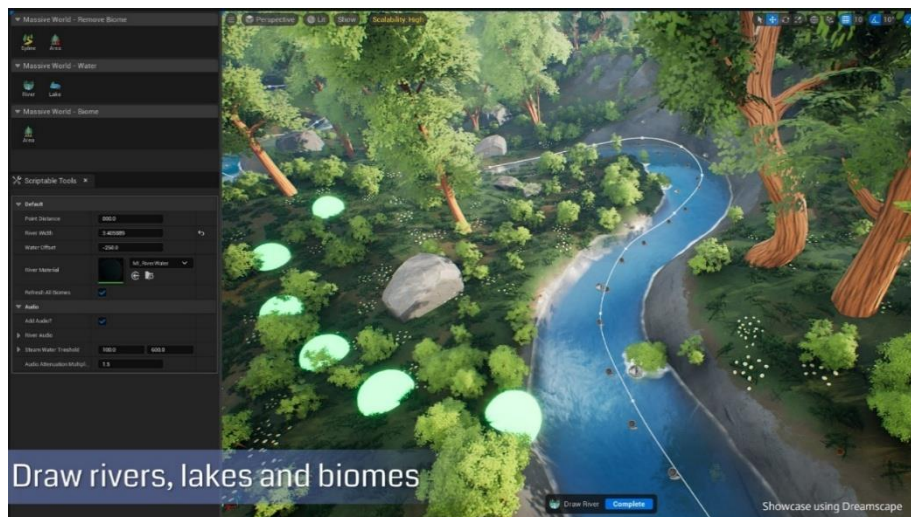
Với sự phát triển vượt bậc của các kỹ thuật trí tuệ nhân tạo và học máy trong thập kỷ gần đây, PCG không còn chỉ dừng lại ở việc tạo bản đồ hoặc môi trường ngẫu nhiên đơn giản, nó có khả năng học từ dữ liệu thực, phân tích hành vi người chơi và sinh ra các kịch bản phù hợp với logic thiết kế, cấp độ thử thách cũng như động lực trải nghiệm của người dùng [45].

1.2 Giới thiệu về kỹ thuật Procedural Content Generation (PCG)

1.1.1. Tổng quan về Procedural Content Generation (PCG)

Kỹ thuật tạo nội dung tự động, còn gọi là Procedural Content Generation (PCG), được định nghĩa là quá trình tạo ra nội dung trò chơi một cách tự động hoặc bán tự động thông qua việc sử dụng các thuật toán, thay vì thiết kế thủ công từng thành phần nội dung bởi con người. Kỹ thuật này đóng vai trò quan trọng trong việc giảm tải khối lượng công việc cho các nhà thiết kế trò chơi, đồng thời mang đến trải nghiệm phong phú, ngẫu nhiên, không lặp lại và có thể cá nhân hóa cho người chơi.

Bên cạnh việc ứng dụng trong việc sinh bản đồ, thiết kế cấu trúc, tạo nhiệm vụ, sinh hình ảnh hay tạo nhân vật, kỹ thuật PCG còn mở rộng ứng dụng ra ngoài phạm vi trò chơi. Cụ thể, PCG đã được nghiên cứu và triển khai trong các lĩnh vực như mô phỏng kiến trúc đô thị, thiết kế nghệ thuật kỹ thuật số, sản xuất phim hoạt hình và đào tạo mô hình trí tuệ nhân tạo



Hình 1 Ứng dụng PCG hỗ trợ tự tạo map trên Unity

1.1.2. Phân loại các kỹ thuật PCG

Các loại kỹ thuật PCG được chia theo thời điểm sử dụng hoặc mức độ can thiệp của

con người, nhưng các kỹ thuật PCG được phân loại chính theo kỹ thuật sinh nội dung

- **Theo kỹ thuật sinh nội dung:**

Kỹ thuật dựa trên luật (Rule-based Generation)

Kỹ thuật này sử dụng các tập luật cố định hoặc các quy tắc hình thức để điều khiển quá trình sinh nội dung. Các luật có thể là:

- **L-systems** (Lindenmayer systems)
- **Context-free grammars (CFGs)**
- **Grammar-based procedural modeling.**

Kỹ thuật dựa trên tìm kiếm (Search-based PCG)

Kỹ thuật này xem nội dung là một bài toán tối ưu, sử dụng các thuật toán tìm kiếm hoặc tiến hóa để tìm ra cấu hình nội dung tốt nhất thỏa mãn hàm mục tiêu.

Các thuật toán phổ biến:

- **Genetic Algorithm (GA):**
- **Simulated Annealing, A*, Monte Carlo Tree Search (MCTS).**
- **Evolution Strategies, Particle Swarm Optimization (PSO).**

Kỹ thuật dựa trên học máy (Machine Learning-based PCG)

Các phần mềm học máy học từ tập dữ liệu mẫu để sinh nội dung tương tự. Chúng có thể là:

- **Supervised Learning:** Học từ cặp dữ liệu đầu vào – đầu ra.
- **Unsupervised Learning:** Như autoencoders để học biểu diễn trừu tượng.
- **Reinforcement Learning (PCGRL)**
- **Deep Generative Models:** Bao gồm GAN (Generative Adversarial Networks), RNN (Recurrent Neural Networks), Transformer.

1.1.3. Khả năng ứng dụng của kỹ thuật PCG

Kỹ thuật PCG được ứng dụng rộng rãi trong nhiều khía cạnh của phát triển trò chơi, bao gồm việc sinh bản đồ ngẫu nhiên, tạo nhiệm vụ, tạo đối thủ hoặc NPC, vật phẩm, cũng như đối thoại tương tác.

Trong lĩnh vực mô phỏng và huấn luyện trí tuệ nhân tạo, PCG giúp tạo ra hàng loạt môi trường huấn luyện đa dạng, từ đó hỗ trợ các mô hình học sâu hoặc các hệ thống xe tự hành tiếp cận được với các kịch bản hiếm gặp, khó tái hiện trong thực tế.

Trong giáo dục và đào tạo, PCG đóng vai trò là công cụ tạo ra các nội dung cá nhân hóa, chẳng hạn như tạo bài tập hoặc câu hỏi kiểm tra phù hợp với năng lực từng học sinh.

Ngoài ra, các trò chơi sử dụng PCG cũng được ứng dụng để mô phỏng các hiện tượng khoa học, phản ứng sinh học hoặc quá trình vật lý trong môi trường học tập.

Bên cạnh đó, trong lĩnh vực kiến trúc và quy hoạch đô thị, PCG được sử dụng để tạo ra các thiết kế tòa nhà, cảnh quan, hoặc các phương án bố trí mặt bằng dựa trên các thông số đầu vào nhất định.

1.1.4. Vai trò của kỹ thuật PCG trong việc phát triển các màn chơi

Trong lĩnh vực phát triển trò chơi điện tử, một màn chơi (level) được hiểu là một đơn vị cấu trúc không gian hoặc logic mà người chơi phải tương tác và vượt qua để đạt được một mục tiêu cụ thể trong tiến trình chơi. Đối với các thể loại trò chơi như platformer, puzzle, roguelike, ... thiết kế màn chơi không chỉ là một yếu tố quan trọng trong việc xây dựng trải nghiệm người dùng, mà còn chiếm một tỷ lệ đáng kể trong tổng chi phí phát triển trò chơi [1].

Trước thực trạng này, kỹ thuật tạo nội dung tự động (Procedural Content Generation – PCG) đã và đang được ứng dụng ngày càng rộng rãi như một giải pháp tiềm năng nhằm tự động hóa quá trình thiết kế màn chơi. Thông qua việc sử dụng tập hợp các quy tắc, thuật toán, hoặc mô hình học máy, PCG có khả năng tạo ra hàng trăm đến hàng nghìn màn chơi khác nhau với chi phí thiết kế thủ công tối thiểu, đồng thời vẫn duy trì được tính nhất quán về logic và thẩm mỹ trong trò chơi tổng thể.

Việc tích hợp PCG trong thiết kế màn chơi mang lại nhiều lợi ích nổi bật như:

- Tăng khả năng chơi lại (replayability) nhờ nội dung ngẫu nhiên, không lặp lại;
- Cá nhân hóa trải nghiệm chơi dựa trên đặc điểm và năng lực của từng người dùng;
- Tạo môi trường học tập phong phú cho cả người chơi lẫn các hệ thống trí tuệ nhân tạo trong trò chơi học thích nghi (*adaptive learning games*).

Mỗi cấp độ đều yêu cầu thuật toán và chiến lược kiểm soát riêng biệt để đảm bảo tính hợp lệ, tính khả thi và giá trị thẩm mỹ trong trải nghiệm người chơi.

1.3 Nghiên cứu các thuật toán PCG sử dụng trong phát triển tự động màn chơi cho game

Việc tạo màn chơi bằng PCG có thể sử dụng một hoặc nhiều kỹ thuật đã được giới thiệu trong phần 1.1. Các thuật toán PCG cho màn chơi có thể được phân thành các nhóm chính như sau:

Thuật toán dựa trên quy tắc (Rule-based): Thuật toán này sử dụng một tập hợp các quy tắc xác định trước (predefined rules) để điều khiển quá trình tạo ra nội dung trò chơi, bao gồm bản đồ, vật thể, vị trí mục tiêu hoặc cấu trúc nhiệm vụ.

Nguyên lý hoạt động cơ bản của Rule-Based Generation là định nghĩa một tập hợp các luật (rules) mô tả cách các phần tử trong game nên được phân bố hoặc tổ chức.

Thuật toán tìm kiếm và tối ưu hóa (Search-based): Thuật toán này coi bài toán sinh nội dung – cụ thể là bản đồ hoặc cấu trúc của một màn chơi – là một quá trình tối ưu hóa.

Thuật toán dựa trên ràng buộc (Constraint-based): Thuật toán này sử dụng các tập hợp ràng buộc (constraints) – tức là các quy tắc, điều kiện hoặc giới hạn được xác định trước – để định hướng quá trình sinh nội dung một cách hợp lệ, có kiểm soát.

Thuật toán dựa trên học máy (Machine Learning-based): Thuật toán sẽ sinh nội dung trò chơi bằng cách sử dụng các mô hình học máy được huấn luyện từ tập dữ liệu mẫu (dataset).

1.4 Nghiên cứu về lý thuyết xác suất, lý thuyết trò chơi

1.4.1 Tổng quan về lý thuyết xác suất

Lý thuyết xác suất (Probability Theory) là một ngành toán học chuyên nghiên cứu các hiện tượng ngẫu nhiên, trong đó kết quả của một quá trình không thể dự đoán một cách chắc chắn, mà thay vào đó được mô tả thông qua các khả năng xảy ra với những mức độ xác suất khác nhau. Đây là nền tảng lý thuyết cốt lõi của nhiều lĩnh vực hiện đại như thống kê suy diễn, học máy (machine learning), trí tuệ nhân tạo (AI), xử lý tín hiệu số, cũng như trong thiết kế các hệ thống tạo nội dung tự động (procedural content generation).

Các biến ngẫu nhiên trong thực tế thường được mô hình hóa bằng các phân phối xác suất, tùy theo tính chất rời rạc hay liên tục của chúng:

- Phân phối rời rạc: bao gồm phân phối Bernoulli, Binomial và Poisson – phù hợp với các hiện tượng đếm số lần xảy ra của một sự kiện;
- Phân phối liên tục: bao gồm Uniform, Normal (Gaussian), và Exponential – thường dùng trong các bài toán liên quan đến thời gian, khoảng cách, hoặc tốc độ;[28]
- Phân phối có điều kiện: đóng vai trò quan trọng trong các hệ thống học xác suất và mô hình hóa quan hệ nhân quả, đặc biệt trong mạng Bayes[28].

1.4.2 Tổng quan về lý thuyết trò chơi

Lý thuyết trò chơi (Game Theory) là một lĩnh vực liên ngành trong toán học ứng dụng, kinh tế học và khoa học máy tính, chuyên nghiên cứu các mô hình tương tác chiến lược giữa các tác nhân có lý trí (rational agents). Mục tiêu chính của lý thuyết trò chơi là xác định các chiến lược tối ưu cho mỗi tác nhân trong một trò chơi, dựa trên giả định rằng tất cả các bên đều hành xử một cách hợp lý và có hiểu biết đầy đủ về quy tắc của trò chơi [18]

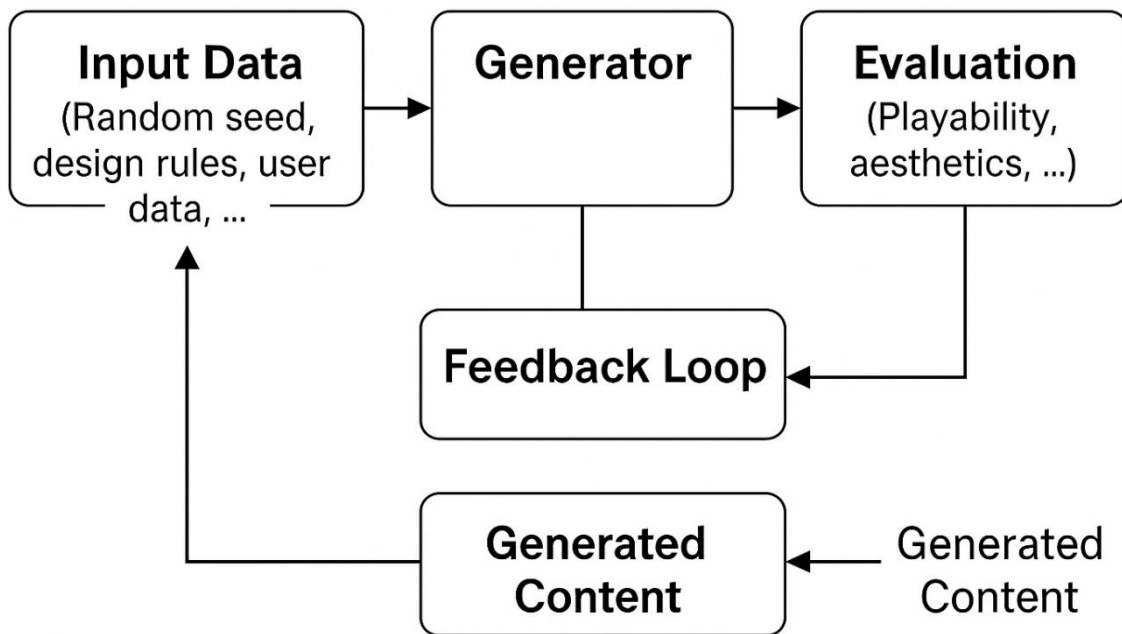
Lý thuyết trò chơi (Game Theory) là một lĩnh vực liên ngành trong toán học ứng dụng, kinh tế học và khoa học máy tính, chuyên nghiên cứu các mô hình tương tác chiến lược giữa các tác nhân có lý trí (rational agents). Lý thuyết này cung cấp một khung lý thuyết toàn diện nhằm phân tích các tình huống trong đó hành động của một đối tượng không chỉ phụ thuộc vào quyết định của chính nó mà còn phụ thuộc vào lựa chọn của các đối tượng khác trong cùng môi trường. Mục tiêu chính của lý thuyết trò chơi là xác định các chiến lược tối ưu cho mỗi tác nhân trong một trò chơi, dựa trên giả định rằng tất cả các bên đều hành xử một cách hợp lý và có hiểu biết đầy đủ về quy tắc của trò chơi [18],

Một khái niệm quan trọng trong lý thuyết trò chơi là cân bằng Nash (Nash Equilibrium) – tình huống trong đó không người chơi nào có động lực thay đổi chiến lược của mình đơn phương vì họ đã đạt lợi ích tối đa, giả sử các người chơi còn lại không thay đổi lựa chọn của mình [19].

CHƯƠNG 2. NGHIÊN CỨU LỰA CHỌN MÔ HÌNH ỨNG DỤNG KỸ THUẬT PCG TÍCH HỢP VÀO PHÁT TRIỂN TỰ ĐỘNG TẠO RA CÁC MÀN CHƠI CHO GAME

Trong chương này, đề tài sẽ nghiên cứu và phân tích lựa chọn từ các thuật toán trong PCG tìm hiểu ở chương 1 để nghiên cứu xây dựng một mô hình phần mềm ứng dụng kỹ thuật PCG có thể tích hợp hỗ trợ tạo ra các màn chơi tự động, đáp ứng nhu cầu của người chơi và yêu cầu của nhà phát triển.

2.1 Mô hình phần mềm sử dụng kỹ thuật PCG để tạo sinh



Hình 1. 1 Mô hình tổng quan phần mềm ứng dụng kỹ thuật PCG tạo sinh

Một hệ thống phần mềm áp dụng kỹ thuật Procedural Content Generation (PCG) thường được cấu trúc gồm bốn thành phần chính: (1) đầu vào (input parameters), (2) mô-đun sinh nội dung (generation engine), (3) mô-đun đánh giá chất lượng (evaluation module), và (4) đầu ra (generated content).

Thành phần đầu vào đóng vai trò xác định các điều kiện khởi tạo và giới hạn cho quá trình tạo nội dung. Các thông số đầu vào ảnh hưởng trực tiếp đến đặc tính ngẫu nhiên, tính đa dạng và mức độ cá nhân hóa của nội dung được tạo ra.

Mô-đun sinh nội dung là thành phần cốt lõi, chịu trách nhiệm tạo ra các bản đồ,

nhệm vụ, vật phẩm hoặc đối tượng trong trò chơi dựa trên đầu vào đã cung cấp. Việc lựa chọn thuật toán phù hợp phụ thuộc vào loại nội dung cần tạo, mức độ phức tạp của thể giới trò chơi và yêu cầu về tính ngẫu nhiên hoặc tính logic.

Mô-đun đánh giá chất lượng có vai trò đảm bảo rằng nội dung sinh ra đáp ứng các tiêu chí thiết kế nhất định. Đối với các hệ thống chuyên sâu hơn, mô-đun này còn thực hiện kiểm tra chéo với dữ liệu thực tế để phát hiện các trường hợp lỗi hoặc các kịch bản bất khả thi.

Kết quả đầu ra là nội dung trò chơi đã được tạo và kiểm duyệt, có thể được sử dụng trực tiếp hoặc tiếp tục tinh chỉnh thông qua vòng lặp phản hồi. Trong các hệ thống hiện đại

Vòng lặp phản hồi (feedback loop) thường được tích hợp để cải tiến quá trình sinh nội dung dựa trên phản hồi của người chơi hoặc phần mềm mô phỏng. Vòng lặp này cho phép hệ thống học hỏi và tự điều chỉnh nhằm tạo ra các nội dung phù hợp hơn với mục tiêu thiết kế và hành vi người chơi

2.2 Phân tích khả năng ứng dụng các thuật toán PCG tạo màn chơi tự động trong các thể loại game theo cấu trúc màn chơi (level-based game)

Trò chơi theo cấu trúc màn chơi (level-based games) là nhóm game trong đó người chơi tiến triển qua các màn chơi riêng biệt, được thiết kế với độ khó và mục tiêu khác nhau, vì vậy các thể loại game này có thể áp dụng các thuật toán PCG nhằm tạo ra hàng trăm đến hàng nghìn màn chơi khác nhau với chi phí thiết kế thủ công tối thiểu, đồng thời vẫn duy trì được tính nhất quán về logic và thẩm mỹ trong trò chơi tổng thể.

Trong nghiên cứu này, em sẽ phân tích áp dụng thuật toán PCG cho các thể loại game cấu trúc màn chơi, các thể loại game phổ biến nhất của loại game cấu trúc màn chơi là: Roguelike games, Platformer games, Puzzle games

2.2.1 Ứng dụng các thuật toán PCG tạo màn chơi tự động cho thể loại game *Roguelike*

Roguelike là một thể loại trò chơi điện tử có đặc điểm nổi bật là thiết kế màn chơi theo kiểu ngẫu nhiên, hệ thống bản đồ được sinh động mỗi lần chơi, độ khó cao, và cơ chế "permadeath" (chết vĩnh viễn không lưu game). Ưu điểm của trò chơi thể loại này nằm ở khả năng tạo ra trải nghiệm mới mẻ, không lặp lại giữa các lần chơi, từ đó nâng cao giá trị chơi lại (replayability) cho người dùng. Việc áp dụng kỹ thuật PCG vào thể loại game này sẽ được

áp dụng chủ yếu trong thành phần (2) Generation Engine – Bộ máy sinh nội dung nhằm sử dụng các thuật toán phù hợp để sinh ra các bản đồ, hoặc vật phẩm hợp lệ và sinh ra map phù hợp nối tiếp tuyến đường hiện có



Hình 2. 1Game thể loại Roguelike

So sánh ưu điểm và nhược điểm ứng dụng các thuật toán của kỹ thuật PCG với thể loại game Roguelike

Bảng 2.1 So sánh ưu điểm và nhược điểm ứng dụng các thuật toán của kỹ thuật PCG với thể loại game Roguelike

Thuật toán PCG	Ưu điểm	Nhược điểm	Mức độ phù hợp
Rule-Based	Kiểm soát chặt nội dung, dễ mở rộng quy tắc, cấu hình tùy chỉnh linh hoạt	Thiếu tính khám phá cao nếu không kết hợp thêm ngẫu nhiên hóa	Trung bình (phù hợp hơn cho bố cục cơ bản và ràng buộc vị trí)
Search-Based	Tạo bản đồ tối ưu theo nhiều tiêu chí, đảm bảo tính khả thi của bản đồ	Chi phí tính toán cao hơn, khó tuning tham số	Trung bình (phù hợp cho tạo đường đi, cấu trúc dungeon phức tạp)
Constraint-Based	Đảm bảo các điều kiện logic/pháp lý nghiêm ngặt	Thiếu khả năng sáng tạo nếu ràng buộc quá cứng	Trung bình–Cao
Machine Learning-Based	Học hành vi người chơi, tự cải thiện bản đồ qua thời gian	Yêu cầu dữ liệu huấn luyện lớn, khó kiểm soát nội dung cụ thể	Thấp–Trung bình với game độc lập nhỏ

2.2.2 Ứng dụng các thuật toán PCG tạo màn chơi tự động cho thể loại Platformers game

Platformer là một trong những thể loại game phổ biến và có lịch sử phát triển lâu đời. Đặc trưng cơ bản của platformer là nhân vật người chơi sẽ di chuyển trong một môi trường

được tổ chức theo chiều ngang hoặc chiều dọc, nhảy qua các vật cản, thu thập vật phẩm và né tránh kẻ thù để đạt đến đích. Nổi tiếng nhất của dòng game thể loại này là Super Mario. Với các trò chơi platformer hiện đại, số lượng màn chơi có thể lên đến hàng trăm, với các biến thể về địa hình, bẫy, hành vi kẻ thù và vật phẩm thưởng, vì vậy việc áp dụng kỹ thuật PCG vào thể loại game này sẽ được áp dụng chủ yếu mục tiêu tạo các cấu trúc platform chính như sàn, khoảng cách nhảy, độ cao tầng từ đó tạo ra các map nối tiếp nhau, ghép lại thành chuỗi hợp lý thành các map nối tiếp nhau.



Hình 2. 2Super mario – Game dạng Platformer nổi tiếng

So sánh ưu điểm và nhược điểm ứng dụng các thuật toán của kỹ thuật PCG với thể loại game platformer

Bảng 2.2 So sánh ưu điểm và nhược điểm ứng dụng các thuật toán của kỹ thuật PCG với thể loại game platformer

Thuật toán PCG	Ưu điểm	Nhược điểm	Mức độ phù hợp
Rule-Based	Khả năng sinh nhanh số lượng lớn màn chơi mà vẫn đảm bảo cấu trúc hợp lý	Tính sáng tạo bị giới hạn, hoàn toàn không có sự đổi mới về cấu trúc màn chơi	Thấp (Không phù hợp với dạng game này)
Search-Based	Có thể sinh ra cấu trúc sáng tạo mà vẫn đảm bảo logic	Tốc độ chậm hơn do phải tìm kiếm qua không gian trạng thái	Trung bình (phù hợp cho sáng tạo các cấu trúc màn chơi)
Constraint-Based	Đảm bảo sinh màn chơi hợp lệ theo các ràng buộc	Thiếu khả năng sáng tạo nếu ràng buộc quá cứng	Thấp (Không phù hợp với dạng game này)

Thuật toán PCG	Ưu điểm	Nhược điểm	Mức độ phù hợp
Machine Learning-Based	Học hành vì người chơi, tự cải thiện bản đồ qua thời gian	Yêu cầu dữ liệu huấn luyện lớn, khó kiểm soát nội dung cụ thể	Thấp–Trung bình không phù hợp với game độc lập nhỏ dạng casual Cao với game có tệp khách hàng lớn và dữ liệu đủ nhiều

2.2.3 Ứng dụng các thuật toán PCG tạo màn chơi tự động cho thể loại Puzzle game

Puzzle game là một thể loại trò chơi điện tử đặc trưng bởi yêu cầu người chơi phải vận dụng tư duy logic, khả năng phân tích, chiến lược hoặc ghi nhớ để giải quyết các bài toán được đặt ra trong game. Một điểm chung nổi bật của puzzle game là hệ thống màn chơi (level) thường được xây dựng dưới dạng phân cấp, nơi mỗi màn chơi là một câu đố độc lập hoặc là một phần trong chuỗi các thử thách có liên kết logic với nhau. Vì vậy có thể thấy, kỹ thuật PCG khi ứng dụng vào thể loại game này sẽ được áp dụng chủ yếu mục tiêu tạo các nội dung cho các màn chơi mới sao cho đảm bảo có thể chơi, cấu trúc logic hợp lệ và có thể tăng độ khó theo từng cấp độ màn chơi



Hình 2. 3Game giải đố - Where is my Water?

So sánh ưu điểm và nhược điểm ứng dụng các thuật toán của kỹ thuật PCG với thể loại game Puzzle

Bảng 2.3 So sánh ưu điểm và nhược điểm ứng dụng các thuật toán của kỹ thuật PCG với thể loại game Puzzle

Thuật toán PCG	Ưu điểm	Nhược điểm	Mức độ phù hợp
Rule-Based	Dễ triển khai, kiểm soát chặt chẽ logic game. Dễ tùy biến theo mức độ khó.	Không thích hợp với thiết kế cần nhiều ngẫu nhiên.	Rất cao (Puzzle game đặc biệt phù hợp với thuật toán Rule-Based)
Search-Based	Có thể tối ưu hóa độ khó hoặc mục tiêu cụ thể (số bước giải, độ phức tạp, thời gian hoàn thành).	Thời gian tìm kiếm có thể lâu nếu không có điều kiện tốt	Cao (phù hợp cho tối ưu hóa độ khó các màn chơi, thời gian hoàn thành)
Constraint-Based	Đảm bảo sinh màn chơi hợp lệ theo các ràng buộc	Yêu cầu xác định rõ ràng các ràng buộc.	Cao (Rất phù hợp với dạng game này)
Machine Learning-Based	Học hành vi người chơi, tự cải thiện bản đồ qua thời gian	Yêu cầu dữ liệu huấn luyện lớn, khó kiểm soát nội dung cụ thể	Thấp–Trung bình không phù hợp với game độc lập nhỏ dạng casual Cao với game có tệp khách hàng lớn và dữ liệu đủ nhiều

Có thể thấy, Puzzle games là một trong những thể loại phù hợp nhất để ứng dụng kỹ thuật PCG, đặc biệt là các thuật toán **Rule-Based**, **Constraint-Based**, và khi cần thiết có thể kết hợp **Search-Based** để điều chỉnh độ khó. Mô hình PCG giúp tiết kiệm thời gian thiết kế, đảm bảo chất lượng logic của màn chơi, đồng thời dễ dàng mở rộng quy mô trò chơi lên hàng trăm màn mà vẫn duy trì tính mới mẻ và hấp dẫn.

2.3 Phân tích lựa chọn thể loại game và thuật toán phù hợp áp dụng đề tài

2.3.1 Lựa chọn thể loại game phù hợp trong đề tài

Dựa vào phân tích về khả năng ứng dụng kỹ thuật PCG vào trong các thể loại cũng như về phạm vi của đề tài là các mô hình thuật toán PCG, luận văn lựa chọn trò chơi giải đố (Puzzle Games) là môi trường ứng dụng chính của kỹ thuật PCG, cụ thể là trong việc tự động tạo sinh màn chơi.

Lý do lựa chọn thể loại này bao gồm:

- **Kiểm soát tốt logic màn chơi:** Puzzle game có tính chất ràng buộc logic cao, dễ định nghĩa và kiểm soát thông qua các quy tắc thiết kế cụ thể, phù hợp với các thuật toán PCG có thể kiểm tra tính hợp lệ như Rule-Based hoặc Constraint-Based.
- **Khả năng cá nhân hóa:** Các màn chơi có thể được sinh ra tương ứng với độ khó, hành vi hoặc khả năng suy luận của người chơi, tăng mức độ phù hợp và hấp dẫn.
- **Tối ưu tài nguyên phát triển:** Việc tự động sinh hàng trăm màn chơi giúp giảm tải đáng kể chi phí thiết kế thủ công, đặc biệt đối với game mobile có yêu cầu nội dung liên tục phù hợp với hướng phát triển và hướng nghiên cứu của đề tài.

2.3.2 Phân tích lựa chọn thuật toán phù hợp

Mục tiêu chính của đề tài là nghiên cứu và đánh giá khả năng áp dụng các thuật toán Procedural Content Generation (PCG) trong việc tạo màn chơi tự động, đồng thời triển khai thử nghiệm với một trò chơi cụ thể nhằm kiểm chứng hiệu quả thực tế của thuật toán. Sau quá trình phân tích khả năng ứng dụng các thuật toán PCG tạo màn chơi tự động trong các thể loại game theo cấu trúc màn chơi (level-based game), nhóm tác giả nhận thấy rằng phương pháp dựa trên quy tắc (Rule-Based PCG) tích hợp thuật toán Constraint-Based là giải pháp phù hợp nhất để áp dụng trong giai đoạn đầu phát triển cho thể loại game Puzzle.

Quyết định lựa chọn thuật toán Rule-Based được đưa ra dựa trên một số tiêu chí kỹ thuật và thực tiễn như sau:

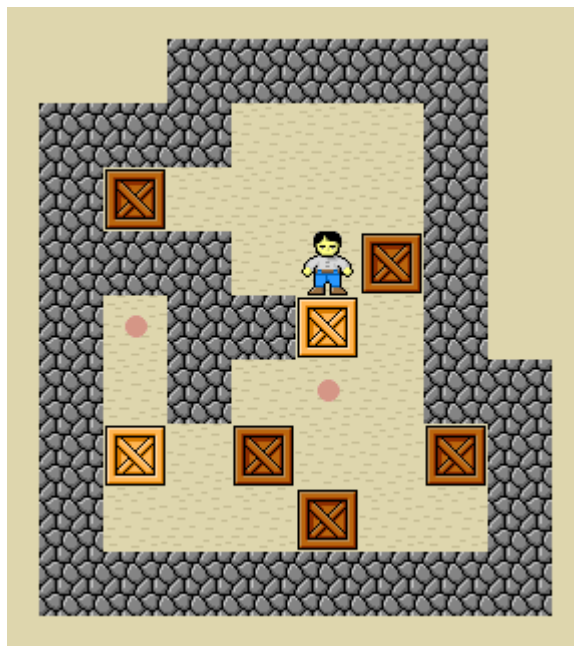
- **Hiệu quả tính toán:** Phương pháp không yêu cầu xử lý khối lượng dữ liệu lớn, không cần huấn luyện mô hình học máy, do đó cho phép thực thi nhanh chóng, tiêu tốn ít tài nguyên hệ thống.
- **Khả năng kiểm soát nội dung:** Rule-Based cho phép các nhà thiết kế trò chơi (game designers) chủ động xác định và điều chỉnh quy tắc sinh nội dung, phù hợp với định hướng thiết kế và mục tiêu trải nghiệm mong muốn của người chơi.
- **Đảm bảo level sinh ra có thể giải được:** Constraint-Based cho phép các nhà thiết kế trò chơi (game designers) chủ động đảm bảo là quy tắc sinh nội dung, phù hợp với định hướng thiết kế và kết quả level sinh ra có thể giải được
- **Tính đơn giản trong triển khai:** Thuật toán có thể dễ dàng được lập trình bằng bất kỳ ngôn ngữ nào mà không yêu cầu thư viện chuyên dụng.
- **Tính tùy biến cao:** Các tập luật (rules) có thể dễ dàng thêm vào, chỉnh sửa hoặc mở rộng tùy theo từng thể loại game, cho phép tái sử dụng linh hoạt và thích nghi với các yêu cầu thiết kế khác nhau

CHƯƠNG 3 . TRIỂN KHAI VÀ XÂY DỰNG PHẦN MỀM

3.1 Giới thiệu game dự kiến tích hợp

Trong quá trình lựa chọn trò chơi mẫu để nghiên cứu và phát triển hệ thống tạo màn chơi tự động sử dụng kỹ thuật Procedural Content Generation (PCG), trò chơi **Sokoban** được lựa chọn nhờ vào tính chất đơn giản về luật chơi và tiềm năng tích hợp các thuật toán sinh nội dung tự động. Sokoban là một trò chơi giải đố cổ điển, trong đó người chơi có nhiệm vụ đẩy các thùng hàng đến đúng vị trí mục tiêu đã định trên bản đồ. Với cơ chế điều khiển cơ bản (chỉ di chuyển và đẩy thùng), trò chơi dễ dàng triển khai bằng các công nghệ web như HTML5, JavaScript và Canvas mà vẫn đảm bảo tính thử thách nhờ không gian chiến lược cao.

Về mặt kỹ thuật, Sokoban rất thích hợp để tích hợp các kỹ thuật PCG, cụ thể là thuật toán Rule-Based kết hợp phân hoạch không gian BSP, nhằm sinh ra các màn chơi có độ phức tạp và khó khăn tăng dần theo từng cấp độ. Đồng thời, hệ thống còn có thể đánh giá khả năng giải được của mỗi màn chơi thông qua các thuật toán tìm kiếm đường đi như BFS, đảm bảo tính khả thi và hợp lý trong thử nghiệm

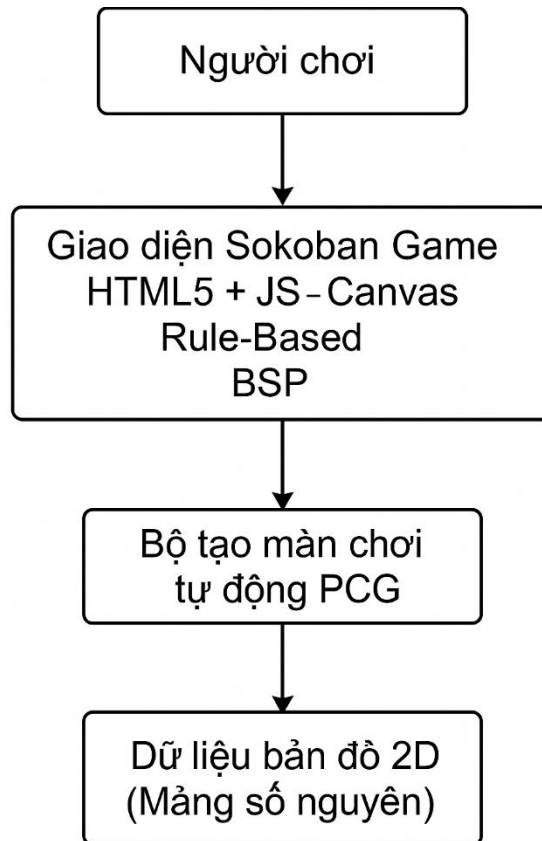


Hình 3.1 Game Sokoban cơ bản

Mục tiêu của chương trình là xây dựng một công cụ sinh màn chơi tự động (level generator) tích hợp trực tiếp vào trò chơi Sokoban, sử dụng kỹ thuật **Procedural Content Generation (PCG)**. Hệ thống cần đảm bảo:

- Các màn chơi sinh ra có thể giải được (solvable).
- Có sự gia tăng về độ khó qua từng cấp độ.

- Cấu trúc bản đồ thay đổi liên tục đa dạng, dựa theo cấu trúc
- Sơ đồ mô hình kiến trúc tích hợp của phần mềm và game



Hình 3.2 Mô hình kiến trúc tích hợp của phần mềm và game

Quy trình hoạt động

- Người chơi khởi động trò chơi và tương tác với giao diện.
- Giao diện thực hiện gọi hàm sinh màn từ bộ tạo màn chơi tự động (PCG Engine).
- Bộ PCG sử dụng thuật toán BSP và Rule-Based để sinh ra dữ liệu bản đồ dạng mảng.
- Giao diện nhận dữ liệu bản đồ 2D, hiển thị trên Canvas và khởi tạo trạng thái game.
- Trong suốt quá trình chơi, nếu người chơi hoàn thành màn, giao diện sẽ gọi lại bộ PCG để sinh level tiếp theo.

3.2 Xây dựng và phát triển phần mềm tích hợp với game

3.2.1 Yêu cầu chức năng

- Sinh màn chơi ngẫu nhiên có kiểm soát: Phần mềm cần đảm bảo khả năng sinh ra các bản đồ màn chơi không lặp lại, nhưng vẫn tuân theo một bộ quy tắc logic để giữ tính nhất quán trong gameplay.
- Tùy chỉnh theo tham số đầu vào: Người phát triển game có thể cung cấp các tham số

- Xác thực tính chơi được (playability): Phần mềm cần đảm bảo rằng mọi màn chơi được sinh ra đều có thể hoàn thành được. Điều này yêu cầu phải có các bước kiểm tra logic:
- Xuất dữ liệu dưới nhiều định dạng: Màn chơi sinh ra cần có thể được lưu trữ và sử dụng ở nhiều môi trường phát triển game khác nhau:
- Tương tác người dùng: Có thể thiết kế thêm giao diện đồ họa cho phép nhà phát triển Xem trực tiếp kết quả màn chơi được sinh.

3.2.2 *Yêu cầu phi chức năng*

- **Hiệu suất:** Thời gian sinh màn chơi phải đảm bảo trong phạm vi 100ms - 1s để đảm bảo không làm gián đoạn dòng chơi hoặc gây trễ trong khởi động màn.
- **Tính ổn định và tin cậy:** Phần mềm không được phép sinh ra màn chơi bị lỗi (crash game, không có mục tiêu, không có người chơi, v.v.).
- **Khả năng mở rộng:** Cho phép tích hợp thêm các thuật toán PCG khác như Search-Based, Constraint-Based trong tương lai mà không cần viết lại toàn bộ phần mềm.
- **Khả năng tích hợp đa nền tảng:** Phần mềm cần hỗ trợ tích hợp vào nền tảng phổ biến hoặc có thể gọi từ server-side để sinh màn chơi động.
- **Bảo trì và nâng cấp:** Codebase rõ ràng, tài liệu đầy đủ, có sẵn cấu trúc module để dễ sửa lỗi và nâng cấp sau này.

3.2.3 *Phân tích lựa chọn công cụ cài đặt hỗ trợ xây dựng phần mềm*

Để hiện thực hóa mục tiêu xây dựng một hệ thống phần mềm hỗ trợ sinh màn chơi tự động dựa trên kỹ thuật Procedural Content Generation (PCG) theo phương pháp Rule-Based, nhóm tác giả đã tiến hành khảo sát, phân tích và lựa chọn một cách có cơ sở các công cụ, ngôn ngữ lập trình và định dạng dữ liệu phù hợp nhằm đảm bảo cả về mặt kỹ thuật lẫn khả năng mở rộng, tích hợp về sau.

Trong quá trình phát triển hệ thống, nhóm đề tài đã lựa chọn bộ công cụ chính bao gồm HTML5, JavaScript và định dạng JSON. Đây là những công nghệ phổ biến, có mức độ tương thích cao với các nền tảng web hiện đại, đồng thời mang lại nhiều lợi thế trong quá trình xây dựng giao diện, xử lý tương tác và mô phỏng trực tiếp nội dung trò chơi. Việc lựa chọn HTML5, JavaScript và JSON không chỉ xuất phát từ tính phổ biến và hỗ trợ cộng đồng rộng rãi, mà còn dựa trên tính mở, linh hoạt, nhẹ và dễ triển khai của các công nghệ này. Bên cạnh đó, các công cụ này cũng hỗ trợ tốt cho mục tiêu triển khai và thử nghiệm phần mềm trực tiếp trên nền tảng Web, cho phép đánh giá khả năng áp dụng thực tế trong môi trường trình duyệt mà không cần cài đặt phức tạp.

3.2.4 *Phát triển phần mềm công cụ sinh màn chơi tự động (level generator) tích hợp trực tiếp vào trò chơi Sokoban*

Load tilemap layout:

- Sử dụng Canvas để vẽ các hình thái mảng 2D.
- Các tile được ánh xạ theo ID:

Chạy PCG logic:

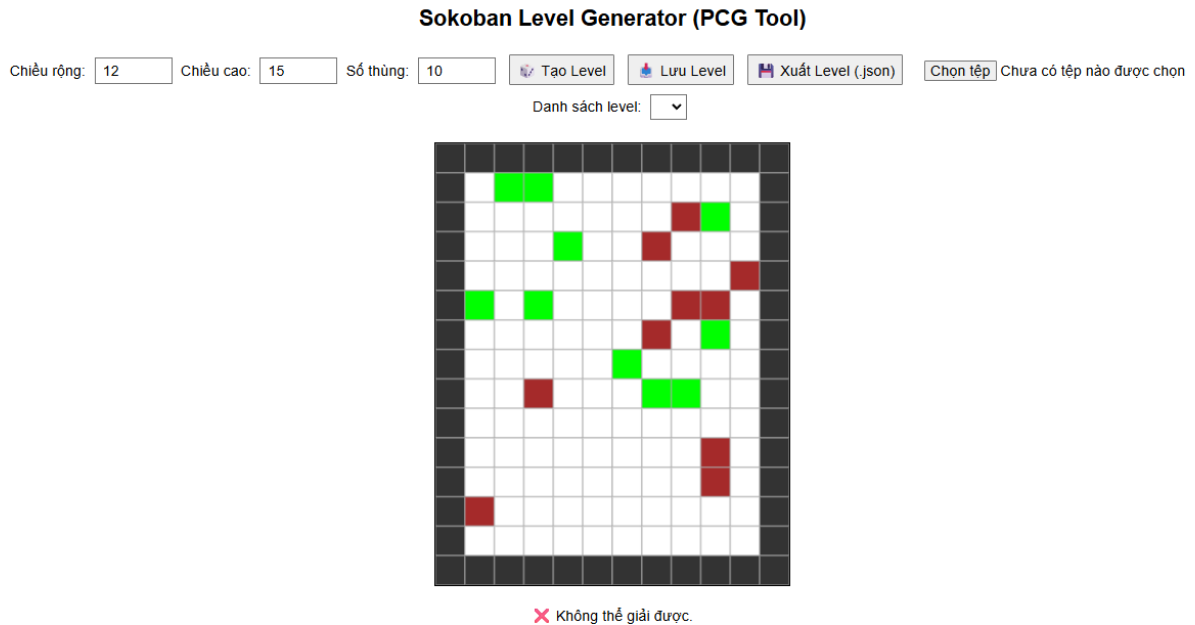
- Phát triển và sử dụng thuật toán BSP (Binary Space Partitioning) trong giai đoạn tiền xử lý để chia không gian bản đồ và sử dụng Rule-based để sinh ra dữ liệu bản đồ theo các mảng.
- Sinh ra bản đồ ngẫu nhiên (map) dựa trên các quy tắc xác định.
- Đảm bảo rằng mỗi màn chơi được tạo ra đều có cấu trúc rõ ràng, khả năng giải được và có độ khó phù hợp với từng cấp độ.
- Khi người dùng nhấn nút “Tạo bản đồ mới”, sự kiện được lắng nghe và gọi MapGenerator.generate().
- Map trả về là một đối tượng JSON chứa thông tin tile, vị trí enemy, trap, player...

Render màn chơi:

- Từ JSON map, hàm createMapFromData() dựng lại level trong GameScene.
- Dùng hàm drawMap để vẽ các đối tượng động.

Đánh giá chất lượng màn chơi:

- Sau khi bản đồ được sinh, người chơi có thể thử chơi trên màn hình trực tiếp.
- Dữ liệu được log lại gồm số lượng trap, enemy, độ dài đường đi từ player đến exit.



Hình 2. 4 Phần mềm PCG tự động tạo ra các màn chơi cho game

3.3 Triển khai, thử nghiệm và đánh giá kết quả

Hệ thống được triển khai và thử nghiệm trong môi trường thực tế để đánh giá hiệu quả. Quá trình này bao gồm cài đặt chạy phần mềm, triển khai chơi thử, hoàn thành các level ghi lại số bước thực hiện cho từng level và kiểm tra các level tự tạo để kiểm tra độ khó của từng level khi tăng dần. Việc triển khai kiểm thử hệ thống đóng vai trò quan trọng trong việc đánh giá chất lượng, độ ổn định và hiệu quả thực tế của phần mềm. Trong phạm vi đề án này, hệ thống được kiểm thử chủ yếu tập trung vào chức năng tạo màn chơi tự động bằng kỹ thuật Procedural Content Generation (PCG) tích hợp vào trò chơi Sokoban. Các chỉ số quan trọng được đánh giá gồm:

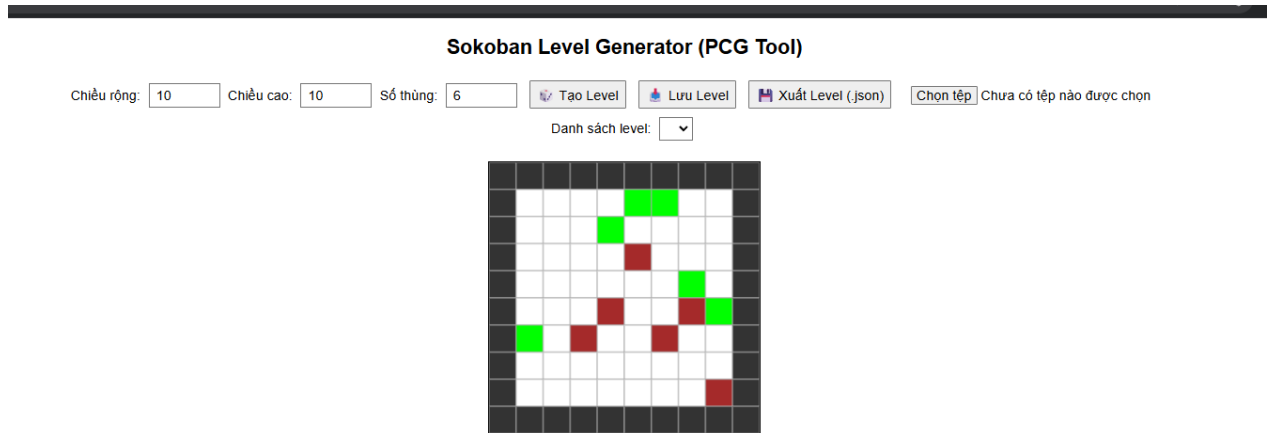
- Tính ổn định và tính khả thi (solvability):
- Hiệu suất xử lý..
- Tính chất tăng dần độ khó:
- Đánh giá trải nghiệm người chơi (User Feedback).

3.3.1 Đánh giá hiệu quả của hệ thống dựa trên kết quả thử nghiệm

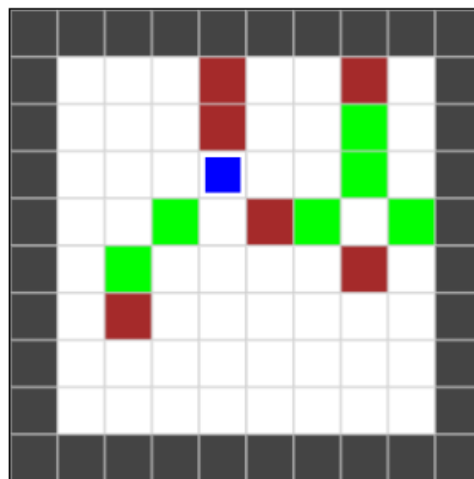
Để phục vụ kiểm thử, hệ thống tiến hành sinh 30 màn chơi tự động bằng cách sử dụng phần mềm hoặc cách tạo Level bằng công cụ với số thùng tăng từ 2 đến 6. Mỗi màn chơi sẽ được đánh giá thông qua:

- Kích thước bản đồ

- Số thùng hàng
- Số mục tiêu
- Độ sâu BSP
- Số bước tối thiểu để hoàn thành (tính bằng thuật toán tìm đường)
- Mức độ đánh giá độ khó (Dễ, Trung bình, Khó)



Hình 3. 3 Level được tạo bằng công cụ



Level 5

Chọn màn chơi: Level 5 ▼

Hình 3.4 Level tự động tạo trong trò chơi

Trò chơi được trải nghiệm bởi nhóm người dùng, kết quả thử nghiệm 30 levels được đánh giá cụ thể như sau:

Bảng 3.1 Kết quả thử nghiệm 30 levels

Nhóm Level	Số lượng thùng (box)	Số bước giải trung bình	Tỉ lệ màn chơi hợp lệ	Tỉ lệ màn chơi giải được	Đánh giá độ khó cảm nhận
1–5	2	8–15 bước	100%	100%	Đễ
6–10	3	12–22 bước	100%	100%	Đễ – Trung bình
11–15	4	20–35 bước	100%	100%	Trung bình
16–20	5	28–45 bước	100%	100%	Khó
21–25	6	35–55 bước	100%	100%	Khó – Rất khó
26–30	6	45–65 bước	96.6% (5/6 màn hợp lệ)	83.3% (5/6 màn giải được)	Rất khó, cần có game sense tốt để giải

Kết quả thử nghiệm chi tiết:

- **Tính ổn định và tính khả thi (solvability):** Hầu hết các màn chơi được tạo ra đều đạt tính hợp lệ cao. Cụ thể, có 29/30 màn chơi hoàn toàn giải được, duy nhất một màn (ở nhóm 21–30) xảy ra hiện tượng deadlock khi 2 thùng bị đặt quá gần tường khiến việc đẩy ra mục tiêu không khả thi.
- **Tính chất tăng dần độ khó:** Mức độ khó được phản ánh rõ nét qua số bước giải trung bình tăng đều. Trong nhóm 1–10, trung bình chỉ cần **10–20 bước** để hoàn thành. Từ nhóm 16–30, số bước dao động từ **30 đến 65 bước**, chứng minh rằng việc tăng số thùng hàng, đồng thời tăng mật độ không gian hẹp, tạo nên độ khó thực tế
- **Hiệu suất xử lý:** Thời gian sinh một bản đồ trung bình chỉ khoảng **30–50 ms**, cho thấy hiệu suất thuật toán rất tốt, phù hợp với môi trường Web game thời gian thực
- **Đánh giá trải nghiệm người chơi (User Feedback):** Mức độ khó tăng dần theo từng level, cấu trúc bản đồ có sự đa dạng, các level sẽ tạo ra các map khác nhau.

Phân tích ưu điểm và nhược điểm của hệ thống

Ưu điểm:

- Hệ thống sử dụng thuật toán PCG, kết hợp giữa phương pháp phân hoạch không gian BSP (Binary Space Partitioning) và kỹ thuật đặt vật thể dựa trên quy tắc (Rule-Based),

cho phép tạo ra các màn chơi mới hoàn toàn mà không cần can thiệp thủ công.

- Nhờ BSP phân chia không gian thành các vùng ngẫu nhiên, bản đồ sinh ra có sự thay đổi đáng kể về bố cục mỗi lần chơi. Cùng với đó, quy tắc tăng dần số lượng thùng hàng và độ sâu phân chia không gian giúp điều khiển độ khó của game tăng dần theo cấp độ.
- Toàn bộ hệ thống được xây dựng bằng HTML5, JavaScript và Canvas, hoạt động tốt trên hầu hết các trình duyệt hiện đại mà không cần cài đặt thêm. Thời gian sinh một màn chơi chỉ khoảng 30–50ms, đảm bảo hiệu suất đủ cao để áp dụng trong môi trường game thực tế và trên các thiết bị cấu hình thấp.
- Kiến trúc phần mềm được tổ chức theo mô-đun rõ ràng, giúp dễ dàng mở rộng số lượng màn chơi, thay đổi quy tắc sinh level, hoặc tích hợp vào các trò chơi khác cùng kiểu lưới.

Nhược điểm:

- Dù hệ thống sinh màn chơi dựa trên các quy tắc hợp lệ, nhưng chưa có cơ chế kiểm tra giải được (solvability) hoặc phát hiện trạng thái bế tắc (deadlock). Trong các màn khó, đã ghi nhận có trường hợp thùng bị kẹt sát tường hoặc góc khiến người chơi không thể hoàn thành màn.
- Hệ thống PCG hiện tại chỉ tăng độ khó tuyến tính thông qua tham số (số thùng, độ sâu BSP). Điều này chưa đủ linh hoạt để điều chỉnh độ khó dựa trên phong cách hoặc hiệu suất chơi thực tế của người dùng.
- Do tập trung vào thuật toán và tính năng lõi, phần giao diện (UI/UX) của hệ thống vẫn còn đơn giản, chưa có hiệu ứng đồ họa nâng cao, âm thanh, hoặc hướng dẫn người chơi rõ ràng.

3.3.2 *Đánh giá khả năng áp dụng hệ thống vào các game FPS khác*

Trong bối cảnh các trò chơi giải đố (Puzzle Game) ngày càng phổ biến và phát triển mạnh mẽ, việc áp dụng một hệ thống hỗ trợ tự động sinh các màn chơi vào các trò chơi này là một bước tiến lớn, không chỉ hỗ trợ nhà phát triển mà còn mở ra nhiều tiềm năng nghiên cứu và ứng dụng trong lĩnh vực trí tuệ nhân tạo (AI). Kỹ thuật tiềm năng ứng dụng rộng rãi của mô hình này đối với nhiều thể loại trò chơi thuộc nhóm puzzle game (trò chơi giải đố).

Lợi thế khi áp dụng vào các trò chơi Puzzle khác

- Các trò chơi như Minesweeper, Bomberman, Tetris dạng phi tuyến, Maze Generator hay các game dạng box-moving puzzles đều có thể tích hợp trực tiếp mô hình sinh nội dung này để tạo ra các màn chơi đa dạng và có độ khó tăng dần.

- Ngoài ra, nếu kết hợp cùng các chiến lược phân tích độ khó thuật toán có thể mở rộng để phục vụ nhu cầu tự điều chỉnh độ khó tạo trải nghiệm chơi cá nhân hóa cho từng người dùng.

Thách thức khi áp dụng vào các trò chơi khác

- Khác biệt luật chơi: Nhiều trò chơi puzzle hiện nay có luật chơi phức tạp, không gian tương tác phi tuyến hoặc yêu cầu tương tác phụ thuộc trạng thái động, khiến cho các quy tắc đặt vật thể tĩnh trở nên không phù hợp.
- Môi trường trò chơi: Với các game dạng puzzle không gian như Portal Puzzle, Escape Room, hay Lemmings, nội dung của màn chơi không đơn thuần là bản đồ 2D và vật thể, mà còn bao gồm cơ chế vật lý, hành vi động, điều kiện thời gian hoặc chuỗi hành động phức tạp.
- Tính đo lường và điều chỉnh độ khó: Việc gán một tham số cụ thể để điều chỉnh độ khó trong giai đoạn sinh nội dung là rất khó nếu không có hệ thống ứng dụng AI để phản hồi từ người dùng hoặc phân tích hành vi người chơi trong thời gian thực

Giải pháp mở rộng và cải tiến hệ thống

- Bổ sung mô-đun phân tích ràng buộc logic (constraint analysis) hoặc kiểm tra giải được tự động (solution validator) sẽ giúp đảm bảo rằng các màn chơi được sinh ra không chỉ hợp lệ về mặt hình thức mà còn khả thi để người chơi hoàn thành
- Kiến trúc của hệ thống cần chuyển đổi từ dạng lưới vuông đơn giản sang dạng đồ thị (graph-based structure) hoặc dạng bản đồ có thể sinh theo nhiều chiều.
- Tích hợp học máy (machine learning) hoặc thuật toán tiến hóa (evolutionary algorithms) để tối ưu hóa quy trình sinh màn chơi, nhằm cá nhân hóa trải nghiệm cho từng người chơi hoặc tạo ra nội dung sáng tạo.

Tiềm năng phát triển trong tương lai

Tiềm năng ứng dụng học máy (machine learning) vào quy trình sinh màn chơi cũng mở ra nhiều cơ hội đổi mới. Việc thu thập dữ liệu hành vi người chơi và sử dụng mô hình học tăng cường (reinforcement learning) hoặc các thuật toán tiến hóa (genetic algorithms) có thể giúp hệ thống tự động điều chỉnh độ khó, tối ưu hoá bố cục màn chơi, đồng thời tạo ra trải nghiệm cá nhân hoá.

KẾT LUẬN

Đồ án tập trung nghiên cứu và triển khai giải pháp tự động hóa tạo màn chơi trong game thông qua kỹ thuật Procedural Content Generation (PCG), nhằm tối ưu quy trình phát triển trò chơi và nâng cao trải nghiệm người chơi. PCG không chỉ giúp tiết kiệm thời gian, công sức mà còn tạo ra nội dung đa dạng, phù hợp với từng người dùng và từng lần chơi.

Trong Chương 1, đồ án đã trình bày tổng quan về khái niệm, vai trò, lý thuyết nền tảng của PCG (bao gồm lý thuyết xác suất và lý thuyết trò chơi), cùng với khảo sát các thuật toán tiêu biểu ứng dụng trong thiết kế nội dung trò chơi.

Chương 2 tập trung vào việc phân tích kỹ thuật và đề xuất hướng tiếp cận phù hợp với mục tiêu đề tài. Từ quá trình đánh giá các tiêu chí như hiệu quả triển khai, khả năng kiểm soát và yêu cầu kỹ thuật, đồ án lựa chọn phương pháp Rule-Based PCG kết hợp với thuật toán phân hoạch không gian BSP để xây dựng hệ thống sinh màn chơi tự động cho trò chơi giải đố (Puzzle Game).

Chương 3 trình bày quá trình triển khai thực tế hệ thống PCG vào một tựa game cụ thể, thu thập dữ liệu và đánh giá hiệu quả. Kết quả cho thấy hệ thống hoạt động hiệu quả, có tính linh hoạt cao, khả năng mở rộng tốt và có thể tùy chỉnh theo yêu cầu nhà phát triển và hành vi người dùng.

Điểm mới của đồ án nằm ở việc:

- Ứng dụng kỹ thuật PCG cho tạo màn chơi puzzle tự động, khác với xu hướng hiện nay chủ yếu dùng PCG để sinh hoạt cảnh hoặc bản đồ 3D trong game.
- Kết hợp linh hoạt giữa thuật toán BSP và phương pháp Rule-Based để định hình bản đồ.
- Kiểm soát tính ngẫu nhiên thông qua tập luật, từ đó tối ưu độ khó và tính hợp lý của màn chơi.

Về định hướng tương lai, đồ án đề xuất ứng dụng kỹ thuật học máy như học tăng cường và thuật toán tiến hóa để thu thập dữ liệu hành vi người chơi, từ đó cá nhân hóa trải nghiệm, tự động điều chỉnh độ khó và tối ưu hóa bố cục màn chơi.

Tổng kết, đồ án đã đạt được các mục tiêu đặt ra từ lý thuyết đến triển khai thực tiễn, chứng minh tính khả thi và hiệu quả của việc tích hợp kỹ thuật PCG vào quy trình phát triển trò chơi hiện đại.