**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

ĐỒ ÁN TỐT NGHIỆP

TRƯỜNG CÔNG NGHỆ THÔNG TIN

**MÔ PHỎNG CHUYỂN ĐỘNG BẦY ĐÀN BẰNG THUẬT TOÁN BOIDS VÀ TỐI ƯU HÓA THUẬT TOÁN**

CBHD: TS. Lê Thị Anh

Sinh viên: Trương Đức Huy

Mã số sinh viên: 2021608438

Hà Nội - 2025

**LỜI CẢM ƠN**

**DANH MỤC HÌNH ẢNH**

**DANH MỤC BẢNG BIỂU**

**MỤC LỤC**

**LỜI MỞ ĐẦU**

1. **Lý do chọn đề tài**
2. **Mục tiêu của đề tài**
3. **Nội dung nghiên cứu**
4. **Bố cục đề tài**

Ngoại trừ các phần lời mở đầu, kết luận và tài liệu tham khảo báo cáo được chia thành bố cục như sau:

* Chương 1: Tổng quan về cơ sở lý thuyết
* Chương 2: …………..
* Chương 3: Cài đặt và đánh giá kết quả

**CHƯƠNG 1: TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT**

1. Tổng quan về thuật toán Boids

Giới thiệu chung

Thuật toán Boids được Craig Reynolds đề xuất năm 1986 nhằm mô phỏng hành vi bầy đàn của các loài chim và cá trong tự nhiên. “Boid” là viết tắt của từ “bird-oid object” (vật thể giống chim).

Thuật toán này không cố gắng lập trình trực tiếp toàn bộ quỹ đạo phức tạp của một đàn chim, mà thay vào đó, mô phỏng dựa trên các quy tắc đơn giản áp dụng cho từng cá thể. Từ những quy tắc cục bộ này, hành vi tập thể phức tạp của cả đàn sẽ tự nhiên xuất hiện.

Boids là một ví dụ điển hình của hiện tượng emergent behavior – khi hành vi toàn cục (toàn đàn) xuất hiện từ các quy tắc đơn giản ở mức cục bộ (mỗi cá thể).

Cách thuật toán Boids hoạt động

Mỗi cá thể trong đàn được gọi là boid và được điều khiển bởi ba quy tắc chính:

* Separation (Tách rời):

Boid giữ khoảng cách nhất định với các boid lân cận để tránh va chạm.

* Alignment (Liên kết hướng):

Boid điều chỉnh hướng đi của mình sao cho gần giống với hướng trung bình của các boid lân cận.

* Cohesion (Kết tụ):

Boid di chuyển về phía trung tâm của các boid lân cận để giữ đàn không bị phân tán

Mỗi boid sẽ có 3 bán kính để xác định các boid lân cận cho từng quy tắc, vector vận tốc và vị trí. Nếu trong từng bán kính không có các boid khác thì boid di chuyển thẳng với vận tốc không đổi. Nếu trong từng bán kính có các boid khác thì các quy tắc trên sẽ được áp dụng và thay đổi vector vận tốc của boid.

* + 1. Quy tắc Separation
* Khái niệm

Separation là quy tắc giúp mỗi cá thể (boid) tránh va chạm với các cá thể khác trong vùng lân cận bằng cách tạo ra một lực đầy có hướng tránh xa các boid khác

* Nguyên tắc hoạt động:
  + Xác định bán kính rS
  + Với mỗi boid, kiểm tra khoảng cách đến các boid xung quanh
  + Nếu boid nào nằm trong bán kính rs, boid đó sẽ tạo ra 1 vector lực đẩy đến boid đang xét, khoảng cách càng gần lực đẩy càng mạnh
  + Tổng hợp tất cả các vector lực đẩy
    1. Quy tắc Alignment
* Khái niệm

Alignment là quy tắc giúp mỗi cá thể (boid) điều chỉnh hướng và vận tốc của mình sao cho tương đồng với các boid lân cận. Điều này tạo ra hiệu ứng cả đàn cùng bay/bơi theo một hướng thống nhất.

* Nguyên tắc hoạt động:
  + Xác định bán kính ra
  + Với mỗi boid, tìm tất cả các boid nằm trong bán kính này.
  + Tổng hợp vector vận tốc của các boid lân cận rồi chia cho số lượng các boid lân cận để lấy vector vận tốc trung bình.
    1. Quy tắc Cohesion
* Khái niệm

Cohesion là quy tắc giúp mỗi cá thể (boid) có xu hướng di chuyển về phía trung tâm đàn để giữ cho đàn không bị phân tán rời rạc.

* Nguyên tắc hoạt động:
  + Xác định bán kính rc
  + Với mỗi boid, tìm tất cả các boid nằm trong bán kính này.
  + Tính vị trí trung bình của nhóm boid lân cận bằng cách tổng hợp vector vị trí của các boid lân cận rồi chia cho số lượng boid lân cận.
  + Boid đang xét tạo ra một vector lực hút hướng về vị trí trung bình đó.

Sau khi áp dụng 3 quy tắc ta được 3 vector lực cho từng quy tắc, tổng hợp 3 lực này sẽ ra lực tác dụng lên boid

A screenshot of a computer screen

AI-generated content may be incorrect.

Có thể dùng thêm các biến trọng số để thay đổi độ lớn từng lực theo ý muốn

Theo định luật II Newton:

= m.

Thuật toán boid không quan tâm đến khối lượng của từng boid nên ta gán m = 1, khi đó vector lực tác dụng lên boid sẽ cung cấp cho boid một vector gia tốc có hướng và độ lớn bằng vector lực.

Vector vận tốc tại thời điểm t:

(t) = 0 + .t

Áp dụng trong mô phỏng máy tính:

new = odd + .

Ứng dụng của thuật toán Boids

Thuật toán Boids được ứng dụng rộng rãi trong nhiều lĩnh vực:

* Đồ họa máy tính & phim ảnh: tạo hiệu ứng đàn chim, đàn cá, đám đông (ví dụ: trong các bộ phim hoạt hình của Disney/Pixar).
* Trí tuệ nhân tạo trong game: điều khiển NPC theo nhóm, mô phỏng đám đông trong game chiến thuật hay sinh tồn.
* Mô phỏng khoa học: nghiên cứu hành vi tập thể của động vật, xe cộ, hoặc con người trong các tình huống khẩn cấp.
* Robot swarm: áp dụng cho nhóm robot tự hành phối hợp với nhau.

Vấn đề hiệu năng trong mô phỏng Boids

Mặc dù các quy tắc của Boids đơn giản, nhưng khi số lượng cá thể lớn (hàng nghìn đến hàng chục nghìn), chi phí tính toán tăng rất nhanh. Mỗi boid cần tính toán khoảng cách và vận tốc tương đối với tất cả các boid khác → độ phức tạp O(n²). Điều này khiến mô phỏng bị chậm khi số lượng cá thể tăng.

Vì vậy, trong đồ án lần này em sẽ sử dụng các hướng tối ưu sau để cải thiện hiệu năng:

* **Spatial Partitioning**: chia không gian bằng **Spatial Hashing**, **Quadtree** để giảm số lượng so sánh.
* **Tận dụng GPU**: **GPU instancing** để tính toán song song.
* **Unity DOTS**: là một framework của unity áp dụng kiến trúc **Entity Component System (ECS)** cùng với **Job System** và **Burst Compiler** để tối ưu bộ nhớ, xử lý đa luồng và tăng tốc độ mô phỏng.