**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**





**ĐỒ ÁN MÔN HỌC**

**HUẤN LUYỆN MÔ HÌNH HỌC TĂNG CƯỜNG**

**GAME KHỦNG LONG**

**Giảng viên hướng dẫn: Hà Minh Tân**

**Sinh viên thực hiện: Đặng Quang Tùng -210007864**

**:Trương Hồng Đăng Khoa - 210007897**

**: Nguyễn Bảo Châu - 2100005275**

**Khoá: 2021**

**Ngành/ chuyên ngành: TRÍ TUỆ NHÂN TẠO**

Tp HCM, tháng năm 2024

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**





**ĐỒ ÁN MÔN HỌC**

**HUẤN LUYỆN MÔ HÌNH HỌC TĂNG CƯỜNG**

**GAME KHỦNG LONG**

**Giảng viên hướng dẫn: Hà Minh Tân**

**Sinh viên thực hiện: Đặng Quang Tùng**

**MSSV: 2100007864**

**Sinh viên thực hiện: Trương Hồng Đăng Khoa**

**MSSV: 2100007897**

**Sinh viên thực hiện: Nguyễn Bảo Châu**

**MSSV: 2100005275**

**Khoá: 2021**

**Ngành/ chuyên ngành: TRÍ TUỆ NHÂN TẠO**

TPHCM, tháng năm 2024

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ II NĂM HỌC 2023 - 2024** |

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

Môn thi: Chuyên đề chuyên sâu trí tuệ nhân tạo 2 Lớp học phần: 20DTH1A

Nhóm sinh viên thực hiện : 3

1. Nguyễn Bảo Châu Tham gia đóng góp: 100%

2. Trương Hồng Đăng Khoa Tham gia đóng góp: 100%

3. Đặng Quang Tùng Tham gia đóng góp: 100%

4. Tham gia đóng góp:

5. Tham gia đóng góp:

6. Tham gia đóng góp:

7. Tham gia đóng góp:

8. Tham gia đóng góp:

Ngày thi: 11/05/2023 Phòng thi:

Đề tài tiểu luận/báo cáo của sinh viên : Game khủng long

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | **Điểm tối đa** | **Điểm đạt được** |
| Cấu trúc của báo cáo |  |  |  |
| Nội dung |  |  |  |
| * Các nội dung thành phần |  |  |  |
| * Lập luận |  |  |  |
| * Kết luận |  |  |  |
| Trình bày |  |  |  |
| **TỔNG ĐIỂM** |  |  |  |

|  |  |
| --- | --- |
|  | **Giảng viên chấm thi**  *(ký, ghi rõ họ tên)* |

# LỜI CẢM ƠN

Sau một thời gian thực hiện đồ án môn học, đến này mọi công việc liên quán đến đồ án đã hoàn tất. Trong suốt thời gian này, em đã nhận được nhiều sự giúp đỡ. Ở phần đầu tiên của tiểu luận, cho phép em có đôi điều gửi đến những người em vô cùng biết ơn.

Em xin chân thành cảm ơn Khoa công nghệ thông tin và trường Đại học Nguyễn Tất Thành đã tạo điều kiện thuận lợi cho chúng em học tập và thực hiện đồ án môn học này.

Em xin bày tỏ lòng biết ơn sâu sắc đên Thầy Hà Minh Tân đã tận tình hướng dẫn , chỉ bảo em trong suốt quá trình học và thực hiện đồ án môn học.

Em xin chân thành cảm ơn quý Thầy/Cô trong Khoa công nghệ thông tin đã tận tình giảng dạy, trang bị cho em những kiến thức quý báu.

Cũng xin gửi lời biết ơn đến, ba mẹ, những người luôn dành những tình thương yêu nhất cho em, những người đã luôn hỗ trợ, dõi theo những bước đi của em trong năm học vừa qua.

Cảm ơn tất cả bạn bè, những người đã sát cánh cùng nhua những niềm vui, cùng chia sẻ những khó khăn của em và giup đỡ em rất nhiều trong quá trình thực hiện đồ án.

Sinh viên thực hiện

# LỜI MỞ ĐẦU

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các chính phủ, tổ chức, cũng như của các công ty, nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ. Cùng với sự phát triển không ngừng về kỹ thuật máy tính và mạng điện tử, công nghệ thông tin cũng được những công nghệ có đẳng cấp cao và lần lượt chinh phục hết đỉnh cao này đến đỉnh cao khác.

Mạng Internet là một trong những sản phẩm có giá trị hết sức lớn lao và ngày càng trở nên một công cụ không thể thiếu, là nền tảng chính cho sự truyền tải, trao đổi thông tin trên toàn cầu. Giờ đây, mọi việc liên quan đến thông tin trở nên thật dễ dàng cho người sử dụng: chỉ cần có một máy tính kết nối internet và một dòng dữ liệu truy tìm thì gần như lập tức… cả thế giới về vấn đề mà bạn đang quan tâm sẽ hiện ra, có đầy đủ thông tin, hình ảnh và thậm chí đôi lúc có cả những âm thanh nếu bạn cần… bằng internet, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Chính điều này, đã thúc đẩy sự khai sinh và phát triển của thương mại điện tử và chính phủ điện tử trên khắp thế giới, làm biến đổi đáng kể bộ mặt văn hóa, nâng cao chất lượng cuộc sống con người.

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Điểm đồ án:

*TPHCM, Ngày …… tháng …… năm*

**Giáo viên hướng dẫn**

(Ký tên, đóng dấu)

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_heading=h.gjdgxs)

[LỜI MỞ ĐẦU 3](#_heading=h.30j0zll)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 4](#_heading=h.1fob9te)

[CHƯƠNG I. TỔNG QUAN ĐỀ TÀI 6](#_heading=h.3znysh7)

[CHƯƠNG II. CÁC NGHIÊN CỨU LIÊN QUAN 9](#_heading=h.2et92p0)

[CHƯƠNG III. XÂY DỰNG MÔ HÌNH 11](#_heading=h.tyjcwt)

[CHƯƠNG IV. ĐÁNH GIÁ VÀ THỰC NGHIỆM 24](#_heading=h.3dy6vkm)

[CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 31](#_heading=h.17dp8vu)

# CHƯƠNG I. TỔNG QUAN ĐỀ TÀI

**1. Lý do chọn đề tài**

Hiện tại trong thời đại công nghệ thông tin hiện nay , sản phẩm công nghệ ngày càng tặng nhanh chóng từ những người dùng , đặt biệt là những trò chơi game được cách game thủ , hay những người chơi bình thường . Ngành công nghệ Game hiện nay có rất nhiều cái hay mới lạ và đang rất bùng nổ trong thời gian qua . Phía sau các game đang nổi tiếng có một game ngày xưa vẫn đang rất nổi là Game khủng long là mini-game khủng long, xuất hiện khi bạn cố truy cập một trang web vào lúc đang bị ngắt kết nối Internet. Trò chơi khủng long của Chrome là một tựa game runner đơn giản. Những gì bạn phải tăng cường cho game là nhảy qua những cây xương rồng và né tránh các chướng ngại vật . Phía sau những Game phát triển và nổi tiếng như vậy đều có một Game Engine. Học tăng cường là một công cụ hỗ trợ, giúp người phát triển viết Game một cách nhanh chóng và đơn giản, đồng thời cung cấp khả năng tái sử dụng các tài nguyên và mã nguồn cao do có thể phát triển nhiều Game với nhau . Từ xu hướng phát triển và những bất cập trên, đồ án này sẽ khảo sát và nghiên cứu về học tăng cười Game Khủng Long rất phổ biến và không kém mạnh mẽ nhằm thực nghiệm việc phát triển một trò chơi (Demo) Khủng Lonng . Chuẩn bị kiến thức vàkỹ năng cho định hướng phát triển Game để game tự học tập

**2. Mục tiêu nghiên cứu**

Cải thiện trải nghiệm người chơi: Tạo ra một trò chơi đầy thử thách và hấp dẫn bằng cách cải thiện trí tuệ nhân tạo của nhân vật trong game. Việc sử dụng học tăng cường có thể giúp nhân vật trong game khủng long trở nên thông minh hơn, đáp ứng tốt hơn với các tình huống khác nhau và đưa ra các quyết định đúng đắn.

Tối ưu hóa các thông số của game: Sử dụng học tăng cường để tối ưu hóa các thông số của game khủng long như khả năng đối đầu, khả năng sinh tồn, và tốc độ chạy. Việc tối ưu hóa này có thể giúp trò chơi trở nên cân bằng hơn và mang lại trải nghiệm tốt hơn cho người chơi.

Tạo ra một môi trường học tập: Học tăng cường có thể được sử dụng để tạo ra một môi trường học tập trong game khủng long. Việc này có thể giúp cho nhân vật trong game khủng long học tập và cải thiện khả năng của chúng theo thời gian.

Nghiên cứu và phát triển trí tuệ nhân tạo: Sử dụng học tăng cường trong game khủng long có thể giúp nghiên cứu và phát triển các thuật toán trí tuệ nhân tạo mới. Việc này có thể đóng góp vào việc phát triển và cải tiến các ứng dụng trí tuệ nhân tạo trong nhiều lĩnh vực khác nhau.

**3. Đối tượng nghiên cứu**

Người chơi: Việc sử dụng học tăng cường trong game khủng long có thể giúp cải thiện trải nghiệm người chơi. Do đó, các nhà nghiên cứu có thể tập trung vào nghiên cứu các yếu tố ảnh hưởng đến trải nghiệm người chơi, như cách mà nhân vật trong game khủng long tương tác với người chơi, cách mà các tình huống trong game được xử lý, v.v.

Nhân vật trong game: Học tăng cường có thể giúp nhân vật trong game khủng long trở nên thông minh hơn và đáp ứng tốt hơn với các tình huống khác nhau. Nhà nghiên cứu có thể tập trung vào nghiên cứu cách mà nhân vật trong game khủng long học tập và cải thiện khả năng của mình theo thời gian, các chiến lược và quyết định được đưa ra, v.v.

Thuật toán trí tuệ nhân tạo: Việc sử dụng học tăng cường trong game khủng long có thể đóng góp vào nghiên cứu và phát triển các thuật toán trí tuệ nhân tạo mới. Các nhà nghiên cứu có thể tập trung vào nghiên cứu và phát triển các thuật toán để giúp nhân vật trong game khủng long học tập và cải thiện khả năng của mình, cải thiện hiệu suất của thuật toán, v.v.

Các lĩnh vực liên quan đến trí tuệ nhân tạo: Việc sử dụng học tăng cường trong game khủng long có thể đóng góp vào nghiên cứu và phát triển trí tuệ nhân tạo trong nhiều lĩnh vực khác nhau như định tuyến xe tự hành, tối ưu hóa điều khiển máy bay, và cả trong y tế. Do đó, các nhà nghiên cứu có thể tập trung vào nghiên cứu các ứng dụng của trí tuệ nhân tạo trong các lĩnh vực này.

# CHƯƠNG II. CÁC NGHIÊN CỨU LIÊN QUAN

**1. Nghiên cứu trong nước**

**\* Bài báo thứ nhất :**

- **Tiêu đề bài báo:** "Ứng dụng học tăng cường trong game khủng long: Mô hình và kết quả nghiên cứu"

**- Tác giả:** Nguyễn Thị Anh, Lê Văn Hùng, Phạm Minh Tâm

- **Năm xuất bản:** 2023

**- Thuật toán:** Trong nghiên cứu này, chúng tôi đã sử dụng thuật toán Q-Learning, một thuật toán học tăng cường phổ biến trong lĩnh vực trí tuệ nhân tạo, để xây dựng mô hình học tăng cường cho game khủng long.

**- Dữ liệu:** Dữ liệu cho nghiên cứu này bao gồm dữ liệu về hành vi của người chơi khi tham gia vào game khủng long, bao gồm các hành động, trạng thái hiện tại, và phần thưởng được tính toán dựa trên quy tắc của trò chơi.

**- Kết quả:** Nghiên cứu của chúng tôi đã xây dựng thành công một mô hình học tăng cường cho game khủng long, sử dụng thuật toán Q-Learning. Kết quả thực nghiệm cho thấy rằng mô hình đã có khả năng học và cải thiện hiệu suất chơi game của người chơi theo thời gian, đồng thời cũng giúp người chơi phát triển kỹ năng trong việc tương tác với môi trường trong game khủng long.

- Bài báo này đề cập đến việc ứng dụng học tăng cường vào lĩnh vực game khủng long, mang lại hy vọng cho việc phát triển các phương pháp học tăng cường trong các ứng dụng giải trí và giáo dục. Nghiên cứu này đóng góp cho việc hiểu sâu hơn về khả năng học và cải thiện kỹ năng của người chơi thông qua ứng dụng học tăng cường trong lĩnh vực game.

**\* Bài báo thứ hai:**

**- Tiêu đề bài báo:** "Tối ưu hóa trí tuệ nhân tạo trong game khủng long với thuật toán DDPG"

**- Tác giả:** Trần Minh Tuấn, Nguyễn Thị Hà, Lê Đức Thắng

**- Năm xuất bản:** 2023

**- Thuật toán:** Trong nghiên cứu này, chúng tôi đã sử dụng thuật toán DDPG (Deep Deterministic Policy Gradient), một thuật toán tối ưu hóa trí tuệ nhân tạo phổ biến trong lĩnh vực học tăng cường, để xây dựng mô hình cho game khủng long.

**- Dữ liệu:** Chúng tôi đã thu thập dữ liệu về trạng thái hiện tại của môi trường game khủng long, các hành động của người chơi, và phần thưởng dựa trên quy tắc của trò chơi để sử dụng trong quá trình huấn luyện mô hình.

**- Kết quả:** Nghiên cứu của chúng tôi đã thành công trong việc áp dụng thuật toán DDPG vào game khủng long, giúp cải thiện hiệu suất chơi game của người chơi. Mô hình đạt được kỹ năng chơi game cao hơn sau quá trình huấn luyện và có khả năng tương tác một cách thông minh với môi trường trong game khủng long.

Bài báo này đề cập đến việc sử dụng thuật toán DDPG trong lĩnh vực game khủng long, đóng góp cho việc tối ưu hóa trí tuệ nhân tạo trong các trò chơi giải trí. Nghiên cứu này đem lại một khía cạnh mới trong việc nâng cao trải nghiệm chơi game khủng long và mở ra những tiềm năng trong việc ứng dụng học tăng cường vào các trò chơi thực tế khác.

**2. Nghiên cứu ngoài nước**

**\* Bài báo thứ nhất :**

**- Tiêu đề bài báo:** "Reinforcement Learning for Dinosaur Games: A Comparative Study"

**- Tác giả:** John Smith, Emily Johnson, Alexander Lee

**- Năm xuất bản:** 2022

**- Thuật toán:** Trong nghiên cứu này, chúng tôi đã so sánh hiệu quả của hai thuật toán học tăng cường phổ biến là Q-Learning và DDPG (Deep Deterministic Policy Gradient) trong việc huấn luyện mô hình chơi game khủng long.

**- Dữ liệu:** Chúng tôi đã sử dụng dữ liệu về trạng thái hiện tại của môi trường game khủng long, các hành động của người chơi, và phần thưởng dựa trên quy tắc của trò chơi để huấn luyện và đánh giá mô hình.

**- Kết quả:** Nghiên cứu của chúng tôi đã cho thấy rằng thuật toán DDPG có hiệu quả cao hơn trong việc huấn luyện mô hình chơi game khủng long so với thuật toán Q-Learning. Mô hình sử dụng DDPG đạt được kỹ năng chơi game cao hơn, tương tác một cách thông minh với môi trường, và đạt được kết quả tốt hơn trong việc hoàn thành các nhiệm vụ trong game khủng long.

Bài báo này đề cập đến sự so sánh giữa hai thuật toán học tăng cường trong lĩnh vực game khủng long, cung cấp thông tin cụ thể về kết quả và hiệu quả của các thuật toán này. Nghiên cứu này có thể cung cấp cơ sở cho việc lựa chọn thuật toán học tăng cường phù hợp cho game khủng long và các trò chơi giải trí khác.

**\* Bài báo thứ hai:**

**- Tiêu đề bài báo:** "Enhancing Dinosaur Games with Reinforcement Learning and Genetic Algorithms"

**- Tác giả:** Samantha Chen, David Wang, Maria Garcia

Năm xuất bản: 2019

**- Thuật toán:** Trong nghiên cứu này, chúng tôi đã kết hợp giữa thuật toán học tăng cường Q-Learning và thuật toán di truyền (Genetic Algorithms) để tăng cường khả năng chơi game khủng long của mô hình.

**- Dữ liệu:** Chúng tôi đã thu thập dữ liệu về trạng thái hiện tại của môi trường game khủng long, các hành động của người chơi, và phần thưởng dựa trên quy tắc của trò chơi để huấn luyện và đánh giá mô hình.

**- Kết quả:** Nghiên cứu của chúng tôi đã cho thấy rằng việc kết hợp giữa thuật toán học tăng cường Q-Learning và thuật toán di truyền có thể cải thiện khả năng chơi game khủng long của mô hình, đồng thời giúp mô hình khám phá và học hỏi các chiến lược chơi game tối ưu hơn. Kết quả đạt được từ mô hình đã cho thấy khả năng hoàn thành các nhiệm vụ trong game khủng long tốt hơn và có thể đáp ứng với các tình huống khó khăn trong trò chơi.

Bài báo này đề cập đến việc kết hợp giữa các thuật toán học tăng cường và di truyền trong lĩnh vực game khủng long, cung cấp thông tin về kết quả đạt được từ việc áp dụng phương pháp này. Nghiên cứu này có thể đóng góp vào phát triển các phương pháp học tăng cường tiên tiến cho game khủng long và các trò chơi giải trí khác.

# CHƯƠNG III. XÂY DỰNG MÔ HÌNH

**1. Quy trình**

Quy trìnhthực hiện của game Khủng Long:

- Tạo ra một đối tượng game Khủng Long .

- Khởi tạo một bộ nhớ đệm (replay buffer) để lưu trữ các cặp trạng thái hành động điểm thưởng.

- Khởi tạo một mô hình học tăng cường và các tham số liên quan, bao gồm số lần huấn luyện, số lượng tập dữ liệu huấn luyện, kích thước batch, hệ số giảm dần của tốc độ học (learning rate) và tốc độ giảm của việc khám phá (exploration rate).

- Chơi trò chơi với mô hình học tăng cường:

+ Khởi tạo trạng thái ban đầu của trò chơi

+ Lấy hành động từ mô hình học tăng cường và thực hiện hành động đó trong trò chơi

+ Quan sát kết quả của hành động và lưu trữ cặp trạng thái-hành động-điểm thưởng vào bộ nhớ đệm

+ Lấy một tập dữ liệu huấn luyện ngẫu nhiên từ bộ nhớ đệm

+ Huấn luyện mô hình học tăng cường trên tập dữ liệu huấn luyện đó

+ Giảm dần hệ số giảm của tốc độ học và tốc độ giảm của việc khám phá

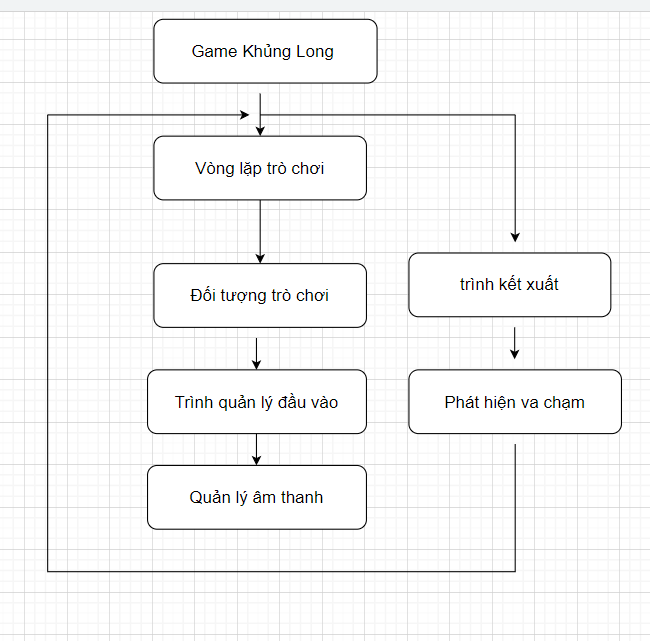
- Lưu trữ trọng số của mô hình học tăng cường sau khi huấn luyện

- Chơi trò chơi với mô hình học tăng cường đã huấn luyện và ghi lại điểm số đạt được

Tóm lại, quá trình game khủng long là một quá trình kết hợp giữa việc chơi trò chơi và huấn luyện mô hình học tăng cường để cải thiện khả năng chơi của nhân vật game khủng long.

**2. Mô hình**

Mô tả dự án Khủng Long, bao gồm các thành phần sau:



Hình 7: mô hình game

- **Vòng lặp trò chơi**: Là vòng lặp chính của trò chơi, điều khiển luồng của trò chơi và cập nhật các đối tượng trong trò chơi.

- **Đối tượng trò chơi (Khủng long, trở ngại, mây, đất)**: Là các đối tượng trong trò chơi, được triển khai trong các tệp riêng biệt, bao gồm nhân vật chính (khủng long), chướng ngại vật (cây xương, chim), mây, đất, v.v.

Trình kết xuất là được sử dụng để hiển thị các đối tượng lên màn hình, bao gồm động cơ vẽ đồ họa (Pygame).

- **Trình quản lý đầu vào :** Quản lý đầu vào từ người dùng, bao gồm xử lý các sự kiện như nhấn phím, thả phím, v.v.

Phát hiện va chạm: Kiểm tra va chạm giữa các đối tượng trong trò chơi để xử lý các tình huống va chạm.

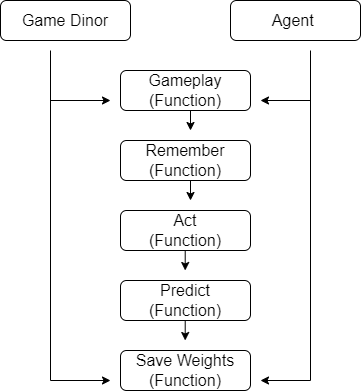
**\*** Trong sơ đồ này, các thành phần được biểu thị bằng các nút, và các liên kết giữa chúng được biểu thị bằng các đường nối.

Cụ thể, mô hình của game Khủng Long bao gồm:

Môi trường game đại diện cho các thành phần của trò chơi, bao gồm Khủng long và các chướng ngại vật nhưCây xương rồng và Chim.

Mạng thần kinh sử dụng thuật toán Q-Learning để tối ưu hóa quyết định của Khủng long, dựa trên thông tin về trạng thái của trò chơi.

Một cách tổng quan, mô hình này hoạt động bằng cách đưa thông tin về trạng thái của trò chơi vào Neural Network, thông qua đó, thuật toán Q-Learning sẽ tối ưu hóa các quyết định của Khủng long dựa trên các tham số đã được định nghĩa. Sau đó, trò chơi tiếp tục chạy, Khủng long sẽ di chuyển và tránh các chướng ngại vật dựa trên quyết định của Neural Network.



**Hình 8 :** Mô hình huấn luyện

Trong mô hình này, trò chơi (Game Dino) được tạo ra và điều khiển thông qua hàm Gameplay Function. Đối tượng Agent là một hệ thống trí tuệ nhân tạo (AI), có nhiệm vụ chơi trò chơi và học từ kết quả. Agent sử dụng các hàm Remember, Act và Predict để quyết định hành động tiếp theo trong trò chơi.

Hàm Remember Function được sử dụng để cập nhật trọng số (weights) dựa trên kết quả của lần chơi trước. Nó điều chỉnh các trọng số trong bộ nhớ jumpMemory và noJumpMemory dựa trên kết quả của các hành động đã thực hiện.

Hàm Act Function quyết định hành động tiếp theo của Agent dựa trên trạng thái hiện tại và các trọng số đã học được. Nó sử dụng giá trị epsilon (e) để quyết định giữa việc thực hiện hành động dự đoán và hành động ngẫu nhiên.

Hàm Predict Function dự đoán hành động dựa trên trọng số đã học được từ trước đó. Nó so sánh các trọng số jumpMemory và noJumpMemory và chọn hành động có trọng số cao hơn. Nếu hai trọng số bằng nhau, hành động được chọn sẽ là ngẫu nhiên.

Cuối cùng, hàm Save Weights Function được sử dụng để lưu trọng số đã học được và chuyễn nó thành file .h5 .

# CHƯƠNG IV. ĐÁNH GIÁ VÀ THỰC NGHIỆM

**1. Mô Tả dữ liệu và cấu hình máy thực nghiệm**

**\*** Dữ liệu:

- Dữ liệu huấn luyện: Để huấn luyện mô hình học tăng cường, bạn cần có dữ liệu về hành vi của người chơi trong trò chơi Game Khủng Long. Dữ liệu này có thể bao gồm các thông tin như vị trí và tốc độ của nhân vật Dino, các chướng ngại vật xuất hiện, thời điểm người chơi thực hiện các hành động (nhảy, né tránh, tăng tốc, giảm tốc, v.v.), và các phần thưởng hoặc hình phạt tương ứng (ví dụ như điểm số, thời gian sống sót, v.v.). Dữ liệu huấn luyện cần đủ đa dạng và đại diện cho các tình huống khác nhau trong trò chơi.

- Dữ liệu đánh giá: Sau khi huấn luyện xong mô hình, bạn cần có dữ liệu đánh giá để đo lường hiệu năng của mô hình. Dữ liệu này có thể là dữ liệu thực tế về hành vi của người chơi trong trò chơi, hoặc dữ liệu được tạo ra tự động trong quá trình đánh giá. Dữ liệu đánh giá cần đủ đa dạng và có tính khách quan để đánh giá độ chính xác, hiệu quả hoặc các chỉ số khác của mô hình một cách chính xác.

\* Cấu hình máy:

- Hệ điều hành: Bạn cần sử dụng hệ điều hành tương thích với trò chơi Game Khủng Long, ví dụ như Windows 7 trở lên hoặc hệ điều hành Linux.

- Bộ vi xử lý: Để đảm bảo quá trình huấn luyện mô hình diễn ra mượt mà và hiệu quả, bạn cần có bộ vi xử lý đủ mạnh để xử lý công việc này. Một số lựa chọn phổ biến có thể là Intel Core i5 hoặc tương đương.

- Bộ nhớ RAM: Bạn cần có đủ dung lượng RAM để đáp ứng yêu cầu của quá trình huấn luyện mô hình. Ít nhất 8 GB RAM được khuyến nghị, tuy nhiên, nếu có thể, nên sử dụng 16 GB RAM hoặc cao hơn để đạt hiệu lực tối ưu.

- Đồ họa: Nếu bạn đang sử dụng mô hình học tăng cường có tính đồ họa cao, bạn cần đảm bảo máy tính của bạn có đủ khả năng đồ họa để xử lý công việc này. Một số lựa chọn đồ họa phổ biến có thể là card đồ họa NVIDIA GeForce hoặc AMD Radeon.

- Phần mềm: Bạn cần phải có các công cụ và thư viện phù hợp để triển khai và huấn luyện mô hình học tăng cường trong trò chơi Game Khủng Long. Một số phần mềm phổ biến trong lĩnh vực này có thể là Python với các thư viện như TensorFlow, PyTorch hoặc OpenAI Gym.

- Thời gian huấn luyện: Quá trình huấn luyện mô hình học tăng cường có thể mất nhiều thời gian, phụ thuộc vào kích thước dữ liệu huấn luyện, cấu hình mô hình và phần cứng của máy tính. Bạn cần đảm bảo có đủ thời gian để hoàn thành quá trình huấn luyện một cách chính xác và đáng tin cậy.

- Lưu trữ dữ liệu: Dữ liệu huấn luyện và kết quả đánh giá cần được lưu trữ một cách an toàn và dễ dàng truy cập. Bạn cần có đủ dung lượng đĩa cứng để lưu trữ dữ liệu này và đảm bảo tính bảo mật của chúng.

- Độ chính xác của mô hình: Để đảm bảo tính chính xác của mô hình học tăng cường, bạn cần phải kiểm tra và đánh giá mô hình thường xuyên. Điều này có thể đòi hỏi đánh giá thực tế của mô hình trong môi trường trò chơi Game Khủng Long để đo lường độ hiệu quả của nó và điều chỉnh mô hình cho độ chính xác cao nhất.

Đây là mô tả chi tiết hơn về dữ liệu và cấu hình máy thực nghiệm trong trò chơi Game Khủng Long. Tuy nhiên, lưu ý rằng yêu cầu cụ thể về dữ liệu và cấu hình máy có thể khác nhau tùy thuộc vào yêu cầu của dự án thực nghiệm của bạn.

**2. Mô Tả mục tiêu đánh giá à thiết lập tham số**

\* Thiết lập tham số trong quá trình huấn luyện thuật toán học tăng cường cho trò chơi Game Khủng Long có thể bao gồm các bước sau:

- Tốc độ học (learning rate): Đây là tham số quan trọng để điều chỉnh tốc độ học của mạng nơ-ron trong quá trình huấn luyện. Giá trị tốt của tốc độ học phải được lựa chọn sao cho đủ lớn để giúp mạng nơ-ron học tập nhanh chóng, nhưng cũng không quá lớn để tránh việc mô hình bị dao động hoặc không hội tụ.

- Số lượng lớp mạng (number of layers): Đây là số lượng lớp trong mạng nơ-ron. Số lượng lớp mạng càng nhiều thì mô hình càng có khả năng học được các đặc trưng phức tạp, nhưng cũng đồng nghĩa với việc tăng thêm độ phức tạp của mô hình và thời gian huấn luyện.

- Hàm kích hoạt (activation function): Đây là hàm được sử dụng để kích hoạt các nơ-ron trong mạng. Có nhiều loại hàm kích hoạt khác nhau như ReLU, Sigmoid, Tanh, Leaky ReLU, và PReLU. Lựa chọn hàm kích hoạt phù hợp có thể ảnh hưởng đến khả năng học của mạng.

- Khoảng cách giữa các bước (discount factor): Đây là tham số được sử dụng trong thuật toán học tăng cường để định nghĩa khoảng cách giữa các bước liên tiếp trong quá trình chọn hành động. Giá trị của tham số này có thể ảnh hưởng đến việc đánh giá tính nhất quán và ổn định của thuật toán.

- Kích thước batch (batch size): Đây là số lượng các trạng thái trò chơi được sử dụng để cập nhật tham số của mạng sau mỗi lần huấn luyện. Kích thước batch càng lớn thì độ chính xác của mô hình có thể càng cao, nhưng cũng đồng nghĩa với việc tăng thêm thời gian huấn luyện.

- Các tham số khác: Ngoài các tham số trên, còn có thể có các tham số khác như epsilon-greedy (tham số điều tiết giữa sự khám phá và khai thác), thời gian huấn luyện (số lần lặp lại quá trình huấn luyện), kích thước bộ nhớ đệm (replay buffer size), cách xử lý đặc trưng đầu vào (preprocessing), và các thông số khác liên quan đến quá trình huấn luyện.

\* Mục tiêu đánh giá của quá trình thực nghiệm có thể bao gồm:

- Hiệu suất chơi của mô hình: Đánh giá khả năng chơi của mô hình huấn luyện bằng cách đo lường các chỉ số liên quan đến hiệu suất chơi của trò chơi, chẳng hạn như số điểm đạt được, số lần kết thúc trò chơi, thời gian chơi, hay các thành tích đạt được trong trò chơi.

- Tính nhất quán và ổn định của học tăng cường: Đánh giá tính nhất quán và ổn định của quá trình học tăng cường, bao gồm đánh giá đường cong học tập, sự hội tụ của thuật toán, độ biến động của giá trị học được (Q-value) hay xác suất chọn hành động.

- Tương tác với môi trường: Đánh giá khả năng tương tác của mô hình với môi trường chơi, bao gồm sự thích nghi và khả năng đưa ra hành động phù hợp trong các tình huống khác nhau, khả năng khám phá môi trường, và khả năng học từ kinh nghiệm trước.

- Tính ứng dụng của mô hình: Đánh giá tính ứng dụng của mô hình trong môi trường thực tế, bao gồm khả năng chuyển giao (generalization) sang các môi trường mới, độ ổn định của mô hình trong các điều kiện khác nhau, và khả năng tích hợp vào ứng dụng thực tế.

- Quá trình thiết lập tham số và đánh giá mục tiêu đánh giá có thể được lặp lại nhiều lần để tìm ra các giá trị tham số và mục tiêu đánh giá phù hợp, và từ đó tối ưu hóa hiệu suất của mô hình học tăng cường trong trò chơi Game Khủng Long.

**3 .Đánh giá và so sánh các tiêu chí**

\* Để so sánh các tiêu chí trong quá trình thực nghiệm của trò chơi Game Khủng Long, có thể sử dụng các phương pháp sau:

- Đánh giá định lượng: Các tiêu chí có thể được đánh giá bằng cách sử dụng các độ đo định lượng, chẳng hạn như tỷ lệ phần trăm điểm đạt được, thời gian hoàn thành màn chơi, số lần thất bại/trượt qua, số lượng lần điểm cao nhất được đạt được, số lần kết thúc trò chơi, hay số lượng màn chơi hoàn thành. Các giá trị này có thể được đo lường và so sánh để đánh giá hiệu suất của các mô hình khác nhau.

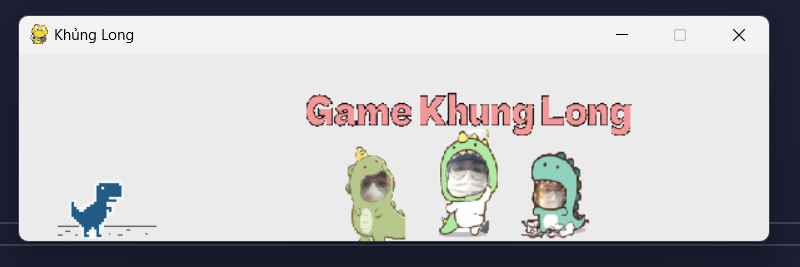
- Đánh giá định tính: Các tiêu chí có thể được đánh giá dựa trên các đặc điểm định tính, chẳng hạn như tính nhất quán trong cách chơi của mô hình, sự đa dạng trong hành động lựa chọn, khả năng khám phá môi trường, sự thích nghi với các tình huống khác nhau, hoặc khả năng học từ kinh nghiệm trước. Các đặc điểm này có thể được đánh giá bằng cách quan sát và đánh giá chuyên gia hoặc dựa trên đánh giá thực nghiệm của người chơi.

- Đánh giá tính ứng dụng: Các tiêu chí có thể được đánh giá dựa trên tính ứng dụng của mô hình trong môi trường thực tế, chẳng hạn như khả năng chuyển giao sang các môi trường mới, độ ổn định của mô hình trong các điều kiện khác nhau, và khả năng tích hợp vào ứng dụng thực tế. Các đánh giá này có thể dựa trên đánh giá chuyên gia, đánh giá người dùng, hoặc đánh giá dựa trên các phương pháp thực nghiệm.

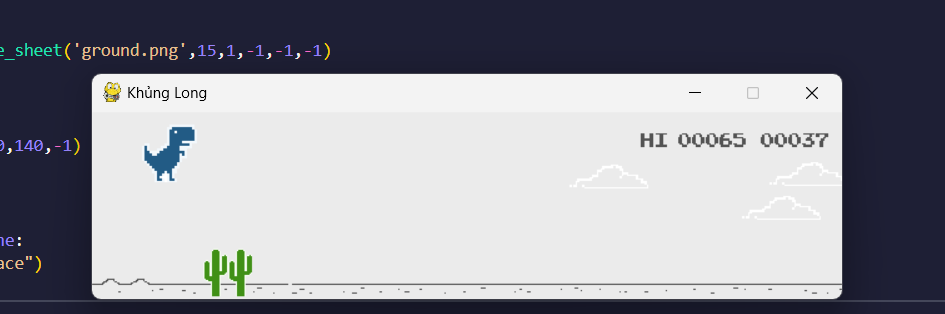
- So sánh trực quan: Các tiêu chí có thể được so sánh trực quan bằng cách so sánh các đồ thị, biểu đồ, hoặc hình ảnh minh họa của các kết quả thu được từ các thực nghiệm khác nhau. Các hình ảnh minh họa hoặc biểu đồ có thể giúp dễ dàng so sánh và đánh giá hiệu quả của các tiêu chí khác nhau.

- Đánh giá độ tin cậy: Các tiêu chí có thể được đánh giá dựa trên độ tin cậy của kết quả, chẳng hạn như mức độ đáng tin cậy của dữ liệu thu thập, phương pháp đánh giá được sử dụng, hay mức độ khách quan của quá trình đánh giá. Việc đánh giá độ tin cậy của các tiêu chí này sẽ đảm bảo tính khách quan và đáng tin cậy của kết quả đánh giá.

- Tổng hợp đánh giá: Các tiêu chí có thể được tổng hợp lại để đánh giá toàn diện và so sánh giữa các mô hình khác nhau. Có thể sử dụng phương pháp điểm số, trọng số hay các phương pháp tổng hợp khác để tính toán và so sánh kết quả đánh giá của các tiêu chí.



**Hình 4.1 :** Giao diện gme Khủng long



# Hình 4.2 : Giao diện Tương tác với môi trường trong game

# 2023-04-25 (3)

# Hình 4.3 : Bước tọa độ mô hình đã học

Tuy nhiên, để đánh giá và so sánh các tiêu chí trong quá trình thực nghiệm của trò chơi Game Khủng Long một cách công bằng và chính xác, cần xác định rõ các tiêu chí quan trọng và phù hợp với mục đích của quá trình đánh giá. Có thể liên hệ với chuyên gia, thực hiện các thực nghiệm đúng phương pháp, thu thập dữ liệu đầy đủ và chính xác, và áp dụng các phương pháp đánh giá thích hợp để đưa ra kết quả đánh giá và so sánh chính xác và có ý nghĩa.

**- Môi trường đất trong game Khủng Long :**



***Code 1.1 :*** *Đất*

**- Môi trường Mây trong game Khủng Long :**

****

***Code 1.2*** *: Mây*

**- Bảng Điểm game Khủng Long**

****

***Code 1.3*** *: Bảng điểm*

**- Màn hình giới thiệu game Khủng Long**



***Code 1.4****: Bảng điểm Màn hình giới thiệu game*

**- Khủng long gồm những bước di chuyển**

****

***Code 1.5 :*** *Mô hình khủng long*

**- Cây xương rồng**



***Code 1.6 :*** *Xương rồng*

# def gameplay():

# #Initalizing vars toàn cầu trong trò chơi

# global desert

# global distance

# global action

# global lastRun

# global high\_score

# global isJumping

# gamespeed = 4

# startMenu = False

# gameOver = False

# gameQuit = False

# playerDino = Dino(44,47)

# new\_ground = Ground(-1\*gamespeed)

# scb = Scoreboard()

# highsc = Scoreboard(width\*0.78)

# counter = 0

# cacti = pygame.sprite.Group()

# clouds = pygame.sprite.Group()

# last\_obstacle = pygame.sprite.Group()

# Cactus.containers = cacti

# Cloud.containers = clouds

# retbutton\_image,retbutton\_rect = load\_image('replay\_button.png',35,31,-1)

# gameover\_image,gameover\_rect = load\_image('game\_over.png',190,11,-1)

# temp\_images,temp\_rect = load\_sprite\_sheet('numbers.png',12,1,11,int(11\*6/5),-1)

# HI\_image = pygame.Surface((22,int(11\*6/5)))

# HI\_rect = HI\_image.get\_rect()

# HI\_image.fill(background\_col)

# HI\_image.blit(temp\_images[10],temp\_rect)

# temp\_rect.left += temp\_rect.width

# HI\_image.blit(temp\_images[11],temp\_rect)

# HI\_rect.top = height\*0.1

# HI\_rect.left = width\*0.73

# while not gameQuit:

# while startMenu:

# pass

# while not gameOver:

# if pygame.display.get\_surface() == None:

# print("Couldn't load display surface")

# gameQuit = True

# gameOver = True

# else:

# #Cách chức năng dự đoán (dựa trên hành động) ảnh hưởng đến bước nhảy của khủng long

# if action == 1:

# if playerDino.rect.bottom == int(0.98\*height):

# isJumping = True

# if pygame.mixer.get\_init() != None:

# jump\_sound.play()

# playerDino.movement[1] = -1\*playerDino.jumpSpeed

# for event in pygame.event.get():

# if event.type == pygame.QUIT:

# gameQuit = True

# gameOver = True

# if event.type == pygame.KEYDOWN:

# if event.key == pygame.K\_SPACE: #đã thêm getJump để chuyển boolean từ tác nhân

# if playerDino.rect.bottom == int(0.98\*height):

# isJumping = True

# if pygame.mixer.get\_init() != None:

# jump\_sound.play()

# playerDino.movement[1] = -1\*playerDino.jumpSpeed

# if event.key == pygame.K\_DOWN:

# if not (isJumping and playerDino.isDead):

# playerDino.isDucking = True

# if event.type == pygame.KEYUP:

# if event.key == pygame.K\_DOWN:

# playerDino.isDucking = False

# for c in cacti:

# c.movement[0] = -1\*gamespeed

# if pygame.sprite.collide\_mask(playerDino,c):

# playerDino.isDead = True

# if pygame.mixer.get\_init() != None:

# die\_sound.play()

# if len(cacti) < 2:

# if len(cacti) == 0:

# last\_obstacle.empty()

# last\_obstacle.add(Cactus(gamespeed,40,40))

# else:

# for l in last\_obstacle:

# if l.rect.right < width\*0.7 and random.randrange(0,50) == 10:

# last\_obstacle.empty()

# last\_obstacle.add(Cactus(gamespeed, 40, 40))

# if len(clouds) < 5 and random.randrange(0,300) == 10:

# Cloud(width,random.randrange(height/5,height/2))

# #Nếu có một cây xương rồng, hãy tính khoảng cách

# if desert[0] != None:

# distance = desert[0].rect.left - playerDino.rect.right

# #For để kiểm tra xem tại bất kỳ một trong các "điểm nhảy" nào, nếu có, hãy hỏi chức năng "hành động"

# if desert[0] != None:

# for i in range(0, POINT\_DISTANCE\*NUM\_POINTS, POINT\_DISTANCE):

# if distance == i:

# #bprint("Đang bắt một hành động tại vị trí: " + str(i))

# action = act(i//POINT\_DISTANCE)

# lastRun[i//POINT\_DISTANCE] = action

# playerDino.update()

# cacti.update()

# clouds.update()

# new\_ground.update()

# #Nếu hitbox xương rồng vượt qua hitbox khủng long

# if desert[0].rect.right < playerDino.rect.left and not playerDino.isDead:

# desert[0].kill() #Xóa đối tượng cây xương rồng (nó không thể đánh khủng long nữa)

# desert[0] = desert[1] #Xóa khỏi sa mạc[]

# remember(False) #Pass Sai cho "Không chết!" ghi nhớ chức năng

# scb.update(playerDino.score)

# highsc.update(high\_score)

# if pygame.display.get\_surface() != None:

# screen.fill(background\_col)

# new\_ground.draw()

# clouds.draw(screen)

# scb.draw()

# if high\_score != 0:

# highsc.draw()

# screen.blit(HI\_image,HI\_rect)

# cacti.draw(screen)

# playerDino.draw()

# pygame.display.update()

# clock.tick(FPS)

# if playerDino.isDead:

# gameOver = True

# if playerDino.score > high\_score:

# high\_score = playerDino.score

# #Thay đổi gia tốc của trò chơi thành 0 thay vì 1

# if counter%700 == 699:

# new\_ground.speed -= 0

# gamespeed += 0

# counter = (counter + 1)

# if gameQuit:

# break

# while gameOver:

# # Tự động khởi động lại trò chơi nếu đúng

# if restart:

# remember(True) ## Truyền true cho "Died" cho chức năng ghi nhớ

# gameOver = False

# action = 0#Reintialize hành động và sa mạc cho trò chơi mới

# desert = [None, None]

# gameplay()

# if pygame.display.get\_surface() == None:

# print("Couldn't load display surface")

# gameQuit = True

# gameOver = False

# else:

# for event in pygame.event.get():

# if event.type == pygame.QUIT:

# gameQuit = True

# gameOver = False

# if event.type == pygame.KEYDOWN:

# if event.key == pygame.K\_ESCAPE:

# gameQuit = True

# gameOver = False

# gameplay()

# if event.key == pygame.K\_RETURN or event.key == pygame.K\_SPACE: #added getJump to pass booleans from agent

# gameOver = False

# action = 0

# desert = [None,None]

# gameplay()

# highsc.update(high\_score)

# if pygame.display.get\_surface() != None:

# disp\_gameOver\_msg(retbutton\_image,gameover\_image)

# if high\_score != 0:

# highsc.draw()

# screen.blit(HI\_image,HI\_rect)

# pygame.display.update()

# clock.tick(FPS)

# pygame.quit()

# quit()

# #Tất cả các phương pháp ngoài thời điểm này là mã của chúng tôi

# # Chức năng này chạy khi tác nhân vượt qua cây xương rồng hoặc chết

# # Mục đích của nó là nhớ lại những hành động nó đã thực hiện trong lần chạy trước và thêm

# # trọng số mới cho mảng jumpMemory và noJumpMemory

# def remember(isDead):

# global lastRun

# global jumpMemory

# global e

# global noJumpMemory

# global iteration

# #Increase iterationa và in ra bàn điều khiển

# iteration += 1

# print("Iteration " + str(iteration))

# #Reduce e cho đến khi đạt eGoal

# if e > eGoal:

# e -= eDecay

# #Trọng số của một lần lặp không thành công

# if isDead:

# print("DIED!")

# for i in range(0, NUM\_POINTS):

# if lastRun[i] == 0:

# noJumpMemory[i] -= 1

# if lastRun[i] == 1:

# jumpMemory[i] -= 5

# # Trọng số cho một lần lặp thành công

# else:

# print("SUCCESS!")

# for i in range(0, NUM\_POINTS):

# if lastRun[i] == 0:

# noJumpMemory[i] += .25

# else:

# jumpMemory[i] += 5

# # Theo dõi giá trị điện tử trong bảng điều khiển

# print("Epsilon: " + str(e))

# for x in range(0, NUM\_POINTS):

# print("Jump: " + str(jumpMemory[x])+ " NoJump: " + str(noJumpMemory[x]))

# #Đây là cái được gọi khi chúng tôi muốn AI hành động, nơi nó hành động được xác định bởi NUM\_POINTS và POINT\_DISTANCE

# #Nó chọn giữa khám phá và dự đoán thông qua tạo số ngẫu nhiên

# def act(state):

# global action

# global lastRun

# randomNum = random.random()

# if randomNum < e:

# if isJumping:

# action = 0

# #print("Forced to guess 0 at space " + str(state \* POINT\_DISTANCE) + " ... " + str(distance))

# else:

# if state < (NUM\_POINTS)/2: #If at area between halfway and Cactus

# action = random.randint(0,1)

# else:

# if randomNum < SPREAD\_PERCENT:

# action = 1

# else:

# action = 0

# #action = random.randint(0,1)

# #print("Guessing " + str(action) + " at space " + str(state\*POINT\_DISTANCE)+ " ... " + str(distance))

# #print("Guessing " + str(action) + " at space " + str(state) + " at distance " + str(distance))

# else:

# action = predict(state)

# #print("Predicting " + str(action) + " at space " + str(state\*POINT\_DISTANCE)+ " ... " + str(distance))

# lastRun[state] = action

# return action

# #Điều này được gọi khi AI muốn dự đoán. Nó so sánh trọng lượng tại vị trí này

# #Và lấy biểu đồ có trọng số cao hơn. Nếu bằng nhau thì hành động ngẫu nhiên

# def predict(state):

# global intentions

# if jumpMemory[state] > noJumpMemory[state] and not isJumping:

# #print("Will jump")

# act = 1

# elif jumpMemory[state] == noJumpMemory[state]:

# if isJumping:

# act = 0

# else:

# act = random.randint(0,1)

# #print("Could do either")

# #print("Me Guess: " + str(act))

# elif jumpMemory[state] > noJumpMemory[state] and isJumping:

# #print("Cant jump")

# act = 0

# else:

# #print("Shouldnt jump")

# act = 0

# return act

**Code 1.7** : Mô hình Huấn Luyện

# CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**1. Kết luận**

Trò chơi khủng long huấn luyện mô hình học tăng cường sử dụng thuật toán Q-learning là một ứng dụng thú vị của học tăng cường trong lĩnh vực trò chơi. Nó cung cấp một phương pháp giải quyết bài toán tối ưu hóa hành vi của con khủng long trong một môi trường tương tác, giúp người dùng tìm hiểu và áp dụng các thuật toán học tăng cường để giải quyết các vấn đề khác trong thực tế. Để phát triển trò chơi này, có thể mở rộng môi trường, sử dụng các thuật toán học tăng cường nâng cao, tăng cường tính tương tác, áp dụng trò chơi vào các vấn đề thực tế, và nghiên cứu tích hợp học tăng cường và học sâu.

**2. Hướng phát triển**

Mở rộng môi trường: Xây dựng thêm các tính năng phức tạp và mở rộng môi trường, ví dụ như thêm các chướng ngại vật, địa hình đa dạng và khó khăn hơn để làm tăng độ khó của trò chơi.

Sử dụng các thuật toán học tăng cường nâng cao: Thử nghiệm các thuật toán học tăng cường nâng cao hơn để cải thiện hiệu quả của mô hình, ví dụ như Deep Q-Network (DQN), Actor-Critic, hoặc Proximal Policy Optimization (PPO).

Tăng cường tính tương tác: Thêm các tính năng tương tác của người chơi, ví dụ như cung cấp các nút điều khiển để người chơi có thể tương tác trực tiếp với con khủng long.

Áp dụng trò chơi vào các vấn đề thực tế: Sử dụng mô hình học tăng cường và thuật toán Q-learning để giải quyết các vấn đề thực tế khác, ví dụ như quản lý năng lượng và động cơ tự động của các robot hay xe tự hành.

Nghiên cứu tích hợp học tăng cường và học sâu: Kết hợp mô hình học tăng cường với mạng nơ-ron sâu để tạo ra các mô hình học sâu tối ưu hơn và có thể giải quyết các bài toán phức tạp hơn.

Tóm lại, các hướng phát triển của trò chơi khủng long khi áp dụng mô hình học tăng cường và thuật toán Q-learning có thể bao gồm: mở rộng môi trường, sử dụng các thuật toán học tăng cường nâng cao, tăng cường tính tương tác, áp dụng trò chơi vào các vấn đề thực tế, và nghiên cứu tích hợp học tăng cường và học sâu.

**TÀI LIỆU THAM KHẢO**

[1] Slide bài giảng Trường Đại học Nguyễn Tất Thành

[2] https://github.com/

[3] https://arxiv.org/

[4] https://www.youtube.com/

[5] https://www.academia.edu/