## **A. CÁC CÔNG NGHỆ SỬ DỤNG TRONG ĐỀ TÀI GAME TANK ONLINE**

### **1. Unity**

**Tổng quan:** Unity là một nền tảng phát triển game đa nền tảng mạnh mẽ, được sử dụng rộng rãi trong công nghiệp game cũng như trong nghiên cứu mô phỏng AI. Unity hỗ trợ cả 2D và 3D, cung cấp môi trường trực quan với khả năng dựng cảnh, ánh sáng, vật lý, và tương tác dễ dàng.

**Vai trò trong đồ án:** Unity là **nền tảng chính để xây dựng game Tank Online**, bao gồm:

* Thiết kế bản đồ, xe tăng, vật thể, vật lý đạn đạo.
* Xử lý va chạm, logic gameplay, camera top-down.
* Cung cấp môi trường mô phỏng cho AI (kết hợp với ML-Agents).

**Kiến thức cần nắm:**

* Scene, GameObject, Prefab, Component.
* Rigidbody, Collider, và hệ thống vật lý.
* Hệ thống sự kiện (Event, Input System).
* Inspector và Editor cơ bản.
* ScriptableObject và quản lý dữ liệu.

**Lợi ích:**

* Tạo môi trường huấn luyện phong phú, có thể mô phỏng vật lý chân thực.
* Kết hợp dễ dàng với ML-Agents để huấn luyện AI hành vi phức tạp.
* Tối ưu hóa cho cả game offline và online.

**Ứng dụng cụ thể:**

* Tank (player & enemy) là các GameObject có Rigidbody và Collider.
* Mỗi Tank có script C# điều khiển di chuyển, xoay nòng, bắn đạn.
* AI Tank chạy trong chế độ “training mode” để ML-Agents thu thập dữ liệu hành động/thưởng/phạt.

### **2. C#**

**Tổng quan:** C# (C Sharp) là ngôn ngữ lập trình hướng đối tượng do Microsoft phát triển, được Unity sử dụng làm ngôn ngữ chính cho các script.

**Vai trò trong đồ án:**

* Xây dựng logic gameplay: điều khiển xe tăng, hệ thống máu, bắn đạn, va chạm.
* Giao tiếp giữa các module: UI, Physics, Animation.
* Gửi/nhận dữ liệu tới Python qua ML-Agents.

**Kiến thức cần nắm:**

* Cấu trúc cơ bản (class, struct, method, property, event).
* Hướng đối tượng: kế thừa, đóng gói, đa hình.
* Xử lý sự kiện (UnityEvent, delegate).
* Coroutine (quản lý thời gian, delay, cooldown).
* Vector3, Quaternion và toán học không gian 3D.

**Lợi ích:**

* Cú pháp rõ ràng, mạnh mẽ, thân thiện với Unity.
* Dễ quản lý project lớn, module hóa dễ dàng.
* Có thể tương tác trực tiếp với thư viện ML-Agents thông qua API.

**Ứng dụng cụ thể:**

* TankMovement.cs: xử lý di chuyển, xoay, tốc độ.
* TankShooting.cs: xử lý việc bắn đạn, cooldown, hướng nòng.
* TankHealth.cs: giảm máu, phát nổ, tính điểm.
* TankAgent.cs: script đặc biệt kế thừa từ Agent (ML-Agents) dùng để định nghĩa hành động & nhận thưởng/phạt.

### **3. Python**

**Tổng quan:** Python là ngôn ngữ lập trình nổi tiếng trong lĩnh vực trí tuệ nhân tạo, học máy và học tăng cường (Reinforcement Learning).

**Vai trò trong đồ án:** Python được sử dụng **bên ngoài Unity** để:

* Huấn luyện mô hình AI thông qua ML-Agents.
* Điều chỉnh tham số huấn luyện (learning rate, gamma, batch size,...).
* Lưu trữ, ghi log và phân tích kết quả huấn luyện (TensorBoard).

**Kiến thức cần nắm:**

* Cách cài đặt môi trường ảo (virtualenv).
* Sử dụng pip, requirements.txt.
* Thư viện AI cơ bản: PyTorch, Numpy, Matplotlib.
* Hiểu quy trình training: observation → action → reward.
* Biết chạy lệnh mlagents-learn và theo dõi TensorBoard.

**Lợi ích:**

* Dễ đọc, dễ mở rộng, cộng đồng lớn.
* Có sẵn nhiều framework hỗ trợ RL.
* Là cầu nối trực tiếp giữa mô hình học máy và Unity.

**Ứng dụng cụ thể:**

* File config.yaml chứa tham số huấn luyện (ví dụ: PPO hoặc SAC).
* Python script tự động chạy training, lưu model.
* Sau khi huấn luyện xong, mô hình .onnx được import ngược lại Unity để Tank AI hoạt động trong game.

### **4. Unity ML-Agents Toolkit**

**Tổng quan:** ML-Agents (Machine Learning Agents) là **gói mở rộng do Unity phát triển**, cho phép sử dụng các thuật toán học tăng cường và học hành vi để huấn luyện nhân vật (agent) trong môi trường ảo.

**Vai trò trong đồ án:**

* Cầu nối giữa Unity và Python.
* Cung cấp môi trường giao tiếp giữa **Tank (Agent)** và **Trainer (Python)**.
* Giúp định nghĩa cách AI thu thập thông tin, hành động, và nhận thưởng.

**Kiến thức cần nắm:**

* Cấu trúc Agent, Behavior Parameters, Decision Requester.
* Các file cấu hình .yaml (chọn thuật toán PPO, SAC, Curiosity, Self-play,...).
* Xuất model và gán lại cho Agent trong Unity.

**Lợi ích:**

* Tích hợp sẵn PyTorch, TensorBoard.
* Hỗ trợ multi-agent training.
* Mô phỏng trực quan (có thể vừa chơi vừa xem AI học).

**Ứng dụng cụ thể:**

* Mỗi Tank Enemy là một Agent.
* Observation: khoảng cách, góc hướng tới mục tiêu, chướng ngại vật.
* Action: xoay trái/phải, di chuyển, bắn.
* Reward: +1 khi bắn trúng, -1 khi bị tiêu diệt

## **B. ỨNG DỤNG REINFORCEMENT LEARNING TRONG GAME TANK ONLINE**

## **I. Giới thiệu tổng quan**

**Reinforcement Learning (Học tăng cường)** là một nhánh của Trí tuệ nhân tạo (AI) thuộc lĩnh vực Học máy (Machine Learning). Khác với học có giám sát (supervised learning) hay học không giám sát (unsupervised learning), học tăng cường dựa trên nguyên tắc **thử - sai (trial and error)** để tìm ra chiến lược hành động tối ưu thông qua **phản hồi phần thưởng (reward)** từ môi trường.

Trong game “Tank Online”, Reinforcement Learning được sử dụng để **huấn luyện AI xe tăng tự hành** có khả năng tự di chuyển, tránh chướng ngại vật, tấn công kẻ địch và sinh tồn trong chiến trường. Việc áp dụng RL vào game giúp AI trở nên thông minh, tự thích nghi và không phụ thuộc vào các kịch bản cứng (hardcoded scripts).

## **II. Cơ sở lý thuyết về Reinforcement Learning**

### **1. Mô hình Markov Decision Process (MDP)**

Reinforcement Learning được mô hình hóa bằng **Markov Decision Process (MDP)**, gồm các thành phần:

* **S (State)**: Trạng thái của môi trường (ví dụ: vị trí xe tăng, hướng địch, vị trí tường,...)
* **A (Action)**: Tập hợp hành động mà agent có thể thực hiện (di chuyển, xoay, bắn,...)
* **R (Reward)**: Phần thưởng nhận được sau mỗi hành động.
* **P (Policy)**: Chiến lược chọn hành động tối ưu.
* **γ (Gamma)**: Hệ số chiết khấu, biểu thị tầm nhìn tương lai của agent.

### **2. Cơ chế học**

Agent sẽ:

1. Quan sát trạng thái môi trường (state).
2. Thực hiện hành động (action).
3. Nhận về phần thưởng (reward) và trạng thái mới.
4. Điều chỉnh chính sách hành động dựa trên tổng phần thưởng nhận được.

Mục tiêu của agent là **tối đa hóa tổng phần thưởng tích lũy** theo thời gian.

## **III. Giới thiệu Unity ML-Agents**

**Unity ML-Agents** là một framework mã nguồn mở do Unity Technologies phát triển, giúp tích hợp các thuật toán Reinforcement Learning trực tiếp vào môi trường Unity.

### **1. Các thành phần chính:**

* **Agent**: Đại diện cho một thực thể thông minh (ví dụ: xe tăng AI).
* **Environment**: Thế giới ảo Unity chứa các agent.
* **Behavior Parameters**: Quy định đầu vào, đầu ra, và cách agent học.
* **Trainer (Python)**: Bộ huấn luyện sử dụng thuật toán RL (chủ yếu là PPO – Proximal Policy Optimization).
* **Policy (.onnx file)**: Mô hình học được sau khi huấn luyện.

### **2. Ưu điểm của ML-Agents:**

* Dễ dàng tích hợp với Unity 3D/2D.
* Hỗ trợ nhiều thuật toán học tăng cường (PPO, SAC,...)
* Tự động xuất mô hình AI dưới dạng file .onnx có thể sử dụng trực tiếp trong game.

## **IV. Quy trình huấn luyện AI Tank bằng Reinforcement Learning**

1. **Xây dựng môi trường Unity**: tạo bản đồ, vật cản, mục tiêu, kẻ địch.
2. **Gắn script Agent** cho xe tăng AI để định nghĩa:
   * Các quan sát cần thu thập (khoảng cách, góc, tốc độ,...)
   * Các hành động có thể thực hiện (xoay, di chuyển, bắn,...)
   * Các quy tắc phần thưởng (trúng đạn, né đạn, va chạm,...)
3. **Cấu hình file YAML** cho quá trình train (số bước, batch size, learning rate,...)
4. **Chạy huấn luyện** bằng lệnh:  
   mlagents-learn config/tank\_config.yaml --run-id=TankAI --train
5. **Quan sát kết quả học** qua TensorBoard hoặc biểu đồ reward.
6. **Xuất policy (.onnx)** và gán lại cho Agent trong Unity để sử dụng khi chơi thật.

## **V. Các khái niệm quan trọng trong Reinforcement Learning**

### **1. Reward Function (Hàm phần thưởng)**

Hàm phần thưởng là yếu tố quan trọng nhất quyết định hành vi của AI. Trong game tank:

* +1 điểm khi bắn trúng kẻ địch.
* -1 điểm khi bị bắn hoặc tự va chạm tường.
* +0.1 điểm khi di chuyển an toàn trong thời gian dài.

### **2. Exploration vs Exploitation**

* **Exploration**: Thử nghiệm hành động mới để khám phá chiến lược tốt hơn.
* **Exploitation**: Áp dụng các hành động đã biết là tốt để tối đa hóa phần thưởng. Agent cần cân bằng hai yếu tố này để học hiệu quả.

### **3. Policy và Value Function**

* **Policy (π)**: Chiến lược chọn hành động tối ưu.
* **Value Function (V)**: Dự đoán phần thưởng tương lai từ trạng thái hiện tại.

## **VI. Ứng dụng Reinforcement Learning vào Game Tank Online**

### **1. Mục tiêu ứng dụng**

* Xây dựng hệ thống AI xe tăng tự hành trong môi trường chiến đấu 2D/3D.
* Giúp AI có khả năng tự ra quyết định và thích nghi mà không cần lập trình cứng.

### **2. Các hành vi học được**

* Tự tìm đường tránh vật cản (Obstacle Avoidance)
* Xác định hướng và khoảng cách mục tiêu
* Ra quyết định tấn công hay phòng thủ
* Học cách né đạn và tìm cover khi cần thiết

### **3. Các thuật toán có thể áp dụng**

* **PPO (Proximal Policy Optimization)** – thuật toán mặc định của ML-Agents.
* **DQN (Deep Q-Network)** – dùng mạng neural dự đoán Q-value.
* **SAC (Soft Actor-Critic)** – dùng trong môi trường liên tục phức tạp.

## **VII. Hướng phát triển**

* Triển khai huấn luyện **Multi-Agent RL** (xe tăng đồng đội và kẻ địch cùng học).
* Nâng cấp thuật toán huấn luyện với **Curriculum Learning** (học theo cấp độ khó tăng dần).
* Tích hợp thêm **Self-Play** để AI học đối kháng với chính nó.
* Tối ưu hiệu suất chạy thật trong Unity (chuyển mô hình RL sang inference).

## **VIII. Tài liệu tham khảo**

1. Unity Technologies – ML-Agents Toolkit Documentation (<https://github.com/Unity-Technologies/ml-agents>)
2. Richard S. Sutton & Andrew G. Barto – *Reinforcement Learning: An Introduction*
3. OpenAI Spinning Up – *Introduction to Reinforcement Learning*
4. Code Monkey – *How to use Machine Learning AI in Unity! (ML-Agents)*