

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



NHẬP MÔN CÔNG NGHỆ THÔNG TIN

Đồ án

GAME 2048

TỰ CHƠI VỚI REINFORCEMENT LEARNING (OPTIONAL)

Giảng viên thực hành: Lê Đức Khoan

Mục lục

1 Chuẩn đầu ra	1
2 Mô tả đồ án	1
2.1 Nội dung chính	1
2.2 Mục tiêu đồ án	1
2.3 Nội dung chi tiết	1
3 Các quy định và yêu cầu	2
4 Yêu cầu mỗi tuần	2
5 Hướng dẫn thực hành	3
5.1 Cấu trúc thư mục	3
5.2 Cài đặt môi trường	5
5.2.1 Cài đặt quản lí môi trường	5
5.2.2 Cài đặt thư viện	5
5.2.3 Vòng lặp game	5



1 Chuẩn đầu ra

Đồ án này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- Tham gia thảo luận, tranh luận theo nhóm trên chủ đề môn học.
- Phân tích, tổng hợp và viết tài liệu kỹ thuật theo mẫu cho trước theo cá nhân hoặc cộng tác nhóm.
- Biết, hiểu thuật ngữ tiếng Anh thuộc các ngành và chuyên ngành.
- Xây dựng một ứng dụng đơn giản, ứng dụng kiến thức ngành Thị giác máy tính.

2 Mô tả đồ án

2.1 Nội dung chính

Xây dựng một game 2048 và kết hợp sử dụng học tăng cường để tự chơi game.

2.2 Mục tiêu đồ án

Sinh viên thực hiện đồ án sẽ nắm được các kiến thức:

- Cấu trúc của game đơn giản (CNPM).
- Các bước xây dựng phần mềm (CNPM).
- Một số công cụ hỗ trợ làm việc trên môi trường Internet (MMT).

Sinh viên thực hiện đồ án sẽ rèn luyện các kỹ năng:

- Lập kế hoạch, làm việc nhóm, tổ chức và quản lý nhóm.
- Tìm hiểu và viết báo cáo.
- Đọc hiểu tiếng Anh và các tài liệu chuyên ngành.

2.3 Nội dung chi tiết

Sinh viên được cung cấp đoạn code xử lý dữ liệu đơn giản. Hãy thực hiện những yêu cầu sau:

1. Cài đặt thành công môi trường lập trình: ngôn ngữ Python, thư viện game Pygame.



2. Chuẩn bị dữ liệu bao gồm: hình ảnh, âm thanh,...
3. Thiết lập setting, vẽ kích thước màn hình chơi game.
4. Game có giao diện: intro, màn chơi.
5. Chức năng: cho người dùng chọn nickname, cho người dùng lựa chọn tiếp tục chơi hay không, thoát khỏi chương trình.
6. Chức năng chơi, luật di chuyển và gộp các ô.

3 Các quy định và yêu cầu

Các kết quả cần đạt được:

- Báo cáo mô tả về đồ án: Mô tả về cách làm, công cụ sử dụng, khó khăn, thuận lợi, các chức năng của chương trình, kế hoạch thực hiện, phân công công việc và đánh giá từng thành viên.
- Github chưa mã nguồn của đồ án. Đảm bảo có phân chia công việc và sử dụng các chức năng của Git như branch, pull request, merge (kiểm tra mỗi tuần về mã nguồn).
- Video báo cáo tiến độ từng tuần.
- Sản phẩm hoàn thiện.

4 Yêu cầu mỗi tuần

- **Tuần 1:** Cài đặt thành công môi trường lập trình, thư viện, chuẩn bị dữ liệu.
- **Tuần 2:** Thiết lập setting, vẽ kích thước bàn chơi game, tạo ngẫu nhiên các số khởi đầu.
- **Tuần 3:** Game có giao diện: intro, màn chơi.
- **Tuần 4:** Chức năng: cho người dùng chọn nickname, cho người dùng lựa chọn tiếp tục chơi hay không, thoát khỏi chương trình.
- **Tuần 5:** Xác định thắng thua, thêm chức năng AI tự chơi (tự chọn).



5 Hướng dẫn thực hành

Dồ án này sẽ sử dụng thư viện Pygame của Python để thiết kế:

5.1 Cấu trúc thư mục

```
game_2048/
  ├── game/
  │   ├── __init__.py
  │   ├── settings.py
  │   ├── app.py
  │   └── core/
  │       ├── __init__.py
  │       └── env_2048.py
  └── rl/
      ├── __init__.py
      ├── agent_dqn.py
      ├── dqn_model.py
      ├── memory.py
      └── train_dqn.py
  └── scenes/
      ├── __init__.py
      ├── intro.py
      └── board.py
  └── main.py
  └── requirements.txt
  └── README.md
```

Giải thích chức năng từng file:

- **game/core/env_2048.py:** Định nghĩa môi trường huấn luyện cho trò chơi 2048 theo phong cách OpenAI Gym. Chứa toàn bộ logic trò chơi: di chuyển, gộp ô và tính toán phần thưởng cho thuật toán học tăng cường.



- **game/rl/dqn_model.py:** Khai báo mô hình mạng nơ-ron QNet (Deep Q-Network) sử dụng thư viện PyTorch. Mô hình nhận đầu vào là trạng thái và xuất ra giá trị Q cho 4 hành động: trên, dưới, trái, phải. Đây là thành phần chính giúp agent học cách ra quyết định tối ưu.
- **game/rl/memory.py:** Triển khai bộ nhớ kinh nghiệm dùng để lưu trữ các mẫu (state, action, reward, next_state, done) trong quá trình huấn luyện. Giúp agent huấn luyện ổn định hơn thông qua việc lấy mẫu ngẫu nhiên để cập nhật trọng số mạng.
- **game/rl/agent_dqn.py:** Chịu trách nhiệm chọn hành động theo chiến lược ε -greedy, cập nhật mạng Q và thực hiện các bước tối ưu hóa (gradient descent). Quản lý việc giảm ε theo thời gian, đồng bộ mạng mục tiêu (target network).
- **game/rl/train_dqn.py:** Tập lệnh huấn luyện chính cho mô hình DQN. Thiết lập tham số (kích thước lưới, số tập, tốc độ học, gamma, v.v.), khởi tạo môi trường, agent và vòng lặp huấn luyện. Sau mỗi tập, nếu điểm số tốt nhất được cải thiện, mô hình được lưu vào thư mục checkpoints/.
- **game/scenes/board.py:** Giao diện trực quan dùng Pygame để hiển thị giao diện chơi game cho người dùng. Kích thước bàn. Bên cạnh đó cũng là giao diện để mô phỏng trò chơi tự chơi với AI.
- **game/scenes/intro.py:** Giao diện trực quan dùng Pygame để hiển thị màn hình chào, nơi người dùng có thể chọn được chế độ chơi là tự chơi hay là AI chơi, nhập thông tin người chơi, ...
- **main.py:** Điểm khởi chạy chính của dự án. Nơi bắt đầu khởi tạo giao diện game và chơi.
- **README.md:** Tài liệu mô tả tổng quan về dự án: hướng dẫn cài đặt môi trường, cách huấn luyện mô hình DQN, cách chạy giao diện hiển thị với Pygame, và cấu trúc thư mục của repository.
- **requirements.txt:** Danh sách các thư viện cần thiết (`torch`, `numpy`, `tqdm`, `pygame`) để cài đặt và chạy chương trình huấn luyện cũng như giao diện hiển thị trò chơi.



5.2 Cài đặt môi trường

5.2.1 Cài đặt quản lý môi trường

Với đồ án này, sinh viên sử dụng các công cụ như conda, miniconda, micromamba để tạo môi trường và cài đặt thư viện. Tham khảo tại: [Micromamba](#).

5.2.2 Cài đặt thư viện

Cài đặt thư viện Pygame. Sau khi đã tạo được môi trường với micromamba.

```
micromamba activate caro_game
pip install pygame
```

5.2.3 Vòng lặp game

Tham khảo tại: [pygame](#).