

Bài tập lớn

Quy chế cộng điểm

- Điểm giữa kì sẽ được chọn nếu bài giữa kì hay bài tập lớn này được điểm cao hơn
- Các bạn có thể làm cả 2, nhưng sẽ chỉ lấy 1 điểm cuối cùng cho điểm bài tập lớn
- Đi vào cái link này ([Book 3.xlsx](#)) (Deadline đi 24-11):
 - Game Sudoku tối đa 2 ng 1 nhóm
 - Decision Tree chỉ được 1 ng

▼ Decision Tree

1. Tìm hiểu về thuật toán **Decision Tree** cho bài toán **Classification**
2. Cài đặt thuật toán **Decision Tree** cho bài toán **classification** bằng **code C++**
 - a. Dữ liệu tập huấn luyện (**trainset**) được cung cấp bao gồm:
 - i. **input_train**: Dữ liệu đầu vào dùng để huấn luyện mô hình.
 - ii. **label_train**: Nhãn tương ứng với **input_train**.
 - b. Dữ liệu tập kiểm tra (**testset**):
 - i. **input_test**: Dữ liệu đầu vào cần dự đoán nhãn.
 - ii. **label_test**: Nhãn thực sự tương ứng với **input_test** (*ẩn đi để chấm điểm*).
 - c. Dữ liệu cần trả về:
 - i. **label_test_pred**: Nhãn dự đoán được từ **input_test** sau khi chạy mô hình.
3. Cài đặt các **criterion** để đánh giá chất lượng phép chia tại mỗi nút của cây:
gini, entropy, ...

4. Tìm hiểu về các chỉ số (**metrics**) để đánh giá kết quả: **Accuracy, Precision, Recall, F1-Score**. Kết quả chấm điểm cuối cùng dựa trên **F1-Score**.
5. Tìm hiểu về **overfitting, underfitting** → cài đặt thêm để các tránh hiện tượng này:
 - a. **max_depth**: Độ sâu tối đa của cây.
 - b. **max_leaf_nodes**: Số lượng lá tối đa của cây.
 - c. **min_samples_split**: Số lượng mẫu tối thiểu để chia một nút.
 - d. **min_samples_leaf**: Số lượng mẫu tối thiểu tại một nút lá.
 - e. **cross_validation**: Áp dụng kỹ thuật chia tập huấn luyện thành nhiều tập con để kiểm tra tính tổng quát của mô hình.
6. Tối ưu mô hình bằng cách **Tune Hyperparameters** tìm bộ tham số cho kết quả tốt nhất

▼ Tạo Trò Chơi Sudoku trong Unity bằng C#

Giới thiệu

Trong dự án này, học sinh sẽ tạo một trò chơi Sudoku cổ điển bằng cách sử dụng Unity và C# scripting. Mục tiêu là hiểu cách tích hợp giữa Unity và C#, cũng như học cách triển khai logic trò chơi và giao diện người dùng.

- Link Github ([Toan-Nguyen26/Student_Sudoku](https://github.com/Toan-Nguyen26/Student_Sudoku))
- Link Drive (Video hướng dẫn + Sudoku.apk cho Nox)
- [Toan Nguyen - Toan Hai Nguyen's Website](#) (Link game của a)

Yêu cầu

- **Phần mềm:**
 - Unity (phiên bản ổn định mới nhất)
 - Visual Studio hoặc bất kỳ IDE nào hỗ trợ C#
 - Plugin Unity-C# để tích hợp

- **Kiến thức:**
 - Hiểu biết cơ bản về Unity và C#
 - Quen thuộc với hệ thống UI của Unity
 - Khái niệm về vòng lặp trò chơi và lập trình hướng sự kiện
- **Đầu ra:**
 - Một file apk chạy được trên 1 android Emulator
 - Game ở dưới dạng 1920 * 1080, không cần phải lo về resolution khác
 - Deadline sẽ là tuần 10, điểm của bài này hoặc bài thi giữa kì cao hơn sẽ được cho vào điểm giữa kì

Các bước thực hiện

1. Thiết lập Dự án

- Tạo một dự án mới trong Unity.
- Thiết lập môi trường: Tạo một Scene mới và tạo canvas cho bảng Sudoku.
- Tích hợp C# với Unity bằng cách sử dụng plugin, đảm bảo cả hai có thể giao tiếp với nhau.

2. Thiết kế Giao diện Người dùng

- Tạo lưới 9×9 bằng hệ thống UI của Unity:
 - Sử dụng các nút (Button) hoặc hộp văn bản (TextBox) để đại diện cho các ô Sudoku.
 - Thiết kế một giao diện đơn giản với các tùy chọn như *Trò Chơi Mới*, *Xóa ô điền*, ...

3. Viết Logic Trò Chơi bằng C#

- Triển khai các chức năng sau trong C#:
 - **Trình tạo Bảng Sudoku:** Một hàm tạo ra một câu đố Sudoku hợp lệ.

- **Thuật toán Giải Đố:** Sử dụng backtracking hoặc một thuật toán hiệu quả khác để giải bảng Sudoku.
- **Hàm Kiểm Tra Tính Hợp Lệ:** Kiểm tra xem đầu vào của người dùng có phải là giải pháp hợp lệ không.
- Đảm bảo mã C# có thể giao tiếp đúng cách với Unity, cập nhật bảng Sudoku theo thời gian thực.

4. Tích hợp Logic C# vào Unity

- Sử dụng các hàm C# để xử lý cơ chế trò chơi:
 - Tạo một câu đố mới khi người chơi nhấn nút *Trò Chơi Mới*.
 - Xóa input người dùng qua tay
 - Chức năng pen để nổi bật khả năng note cho người dùng
 - Hint sẽ đưa ra đáp án đúng cho người chơi
 - Pause sẽ ngưng thời gian của game (Không bắt buộc)

5. Thêm Các Tính Năng Trò Chơi

- Làm nổi bật các mục nhập sai bằng màu đỏ khi người chơi mắc lỗi.
- Thêm tính năng hiển thị gợi ý, trong đó một con số được điền cho người chơi nếu họ yêu cầu.
- Bao gồm hiệu ứng âm thanh khi nhấn nút và khi hoàn thành câu đố (tùy chọn).
- Thêm đồng hồ đếm giờ để theo dõi thời gian người chơi giải đố. (tùy chọn).

6. Kiểm Tra Cuối Cùng và Tinh Chỉnh

- Kiểm tra trò chơi để phát hiện lỗi và đảm bảo logic hoạt động chính xác.
- Đảm bảo trò chơi đẹp mắt và dễ sử dụng.

7. Kết quả

- File apk của trò chơi
- Link Github của trò chơi - bao gồm tất cả các thành viên