أسئلة متكررة بخصوص أفكار المشاريع:

- يجب أن يكون هناك طريقة واضحة لإظهار نتائج المشروع، ولإدخال البيانات للـalgorithm .. ولذلك قد يتطلب المشروع Basic GUI .. في هذه الحالة يجب أن يكون هناك Basic GUI لإدخال الإختيارات وإظهار النتائج، وليس على الـGUI أي درجات، ولكن عدم وجوده سيتسبب في خصم درجات.

- الـdevelopment platform هي الـtools والـprogramming languages والـlibraries المستخدمة في المشروع.

- الـdiagrams المطلوبة هي block diagrams وflowcharts توضح الـalgorithms المستخدمة، إلى جانب plots توضح النتائج، إلى جانب أي UML Diagrams تجدونها ضرورية (مثلاً، use-case diagram توضح الـfunctions/features من وجهة نظر المستخدم).

- أفكار المشاريع التي بها puzzles كالـSudoko والـN-Queens والـN-Puzzle، المشروع هو للعثور على حل للعبة بشكل عام (بداية الـpuzzle يجب أن تختلف كل مرة)، وليس لحالة واحدة خاصة فقط.

- أفكار المشاريع التي بها Two-Player Games، المشروع هو لعمل لاعب ذكي واحد فقط لهذه اللعبة، بينما الطرف الآخر من اللعبة هو إنسان وليس آلة. ويجب على اللاعب الذكي استخدام الـapproach المحدد لتقرير كل حركة/خطوة يقوم بها من اللعبة (كرد على كل حركة يقوم بها الطرف الآخر.

- أفكار المشاريع التي بها (heuristic functions)، يجب حل المشكلة باستخدام اثنين heuristic functions على الأقل، ويجب حل المشكلة باستخدام كل heuristic function بشكل منفصل، ثم المقارنة بين تأثير كل منها على سرعة الوصول للحل وجودة الحل.

- أفكار المشاريع التي بها Two-Player Games، يجب تطبيق الـMinimax والـAlpha-Beta Pruning واستخدام الـHeuristics بشكل منفصل (أي تطوير إمكانية تحديد كل حركة في اللعبة، مرة باستخدام الـMinimax فقط دون أي Heuristics "إن كان الـsearch state space" بسيط كالـTic-Tac-Toe والـ8-Puzzle، ثم مرة أخري ولكن مع تطبيق الـAlpha-Beta Pruning، ثم Minimax باستخدام إحدى الـHeuristics، ثم Minimax باستخدام الـ Heuristic Function الثانية).

- في الـTic-Tac-Toe، يجب بالأضافة لما سبق تطبيق الـSymmetry Reduction والـHeuristic Reduction، كلاهما بشكل منفصل.

- أفكار المشاريع التي بها Two-Player Games أو puzzles، إذا لم يكن الـboard size موضح في الـproject description فيجب الإلتزام بالـstandard board size (مثلاً الـIntelligent Go Player يجب تطبيقه على board 19 في 19).

- أفكار المشاريع التي تعتمد على مدخلات مختلفة (Job Scheduling Problem، والـFaculty's Timetable Scheduling Problem، والـ Vehicle Routing Problem، والـKnapsack Problem)، يجب على المستخدم أن يدخل كل أو معظم المدخلات المطلوبة (وليس أن يقوم الـalgorithm كل مرة بحل المشكلة باستخدام نفس الـdata).

- أفكار المشاريع التي بها Two-Player Games أو puzzles لا تحتاج إلى Dataset لأنه لا يوجد بها training phase لأنها تصنف ضمن الـAI ولكن ليس بها ML كما أوضحت سابقاً.

- بالنسبة لأفكار المشاريع التي تحتاج إلى Dataset، بامكانكم استخدام أي dataset أخرى بدلاً من الـdataset المقترحة في الـprojects' description.

- جميع الأفكار التي تحتاج إلى Dataset، عند مناقشة وتجربة المشروع، يتم ذلك باستخدام testing instances من الـdataset المستخدمة، والطلاب غير مطالبين باستخدام data خارجية.

- جميع الأفكار التي تحتاج إلى evolution أو training، أو إلى testing، يجب عمل plots توضح الـevolution أو الـtraining، وجدول يوضح نتائج الـtesting (ويفضل عن طريق الـN Fold Cross Validation كما أوضحت بالمحاضرات).

3030306010469 4

- أفكار المشاريع التي بها تطبيق طريقتين أو أكثر (كـbothGenetic Algorithms & Differential Evolution أو كـGenetic Algorithm AND the Backtracking Algorithm)، يجب حل المشكلة باستخدام كل طريقة منهما بشكل منفصل، ثم المقارنة بين كفاءة كل منهما في سرعة الوصول للحل وجودة الحل.

- أفكار المشاريع التي بها تطبيق Decision Trees & Random Forests، يجب حل المشكلة أولاً باستخدام Decision Tree واحدة فقط، ثم باستخدام الـRandom Forest، فكل طريقة منهما تطبق بشكل منفصل، ثم يتم المقارنة بين كفاءة كل منهما في سرعة الوصول للحل وجودة الحل.

- أفكار المشاريع التي بها Object Detection، المطلوب تصنيف صور بها Object واحد فقط بكل صورة، وليس تصنيف أكثر من Object من صورة واحدة.

- أفكار المشاريع التي بها تطبيق an Artificial Neural Network، يجب حل المشكلة باستخدام ANN تقليدية، ثم إذا رغب الفريق في تطبيق الـCNN كحل إضافي فبإمكانه عمل ذلك بعد تطبيق المطلوب.

أسئلة بخصوص يوم المناقشة:

- يجب طباعة الـreport وتسليمه كـHard Copy عليه أسماء كل أعضاء الفريق، ورقم كل طالب، والمستوى الدراسي والقسم/الشعبة لكل عضو.

- يجب في الـreport وضع رابط لـshared folder عليه كل الـcode والـreport (بـShortened URL) .. ويجب أن يكون الـfolder متاح حتى موعد ظهور النتائج لنتمكن من الرجوع إليه في أي وقت.

N.B.; A documentation (report) should accompany the project. Provide clear and concise (yet, comprehensive) documentation explaining the functionality, algorithms used, and instructions for users to interact with the application. The report should explain the AI algorithms implemented, any design choices, the design rationale behind heuristic functions (if utilized), the rationale behind the values of any parameters/hyper-parameters in the employed algorithms. The documentation should serve as a guide for users and developers to understand the project's inner workings.

نموذج يمكن إتباعه عند كتابة الـreport:

Introduction and Overview:

- Project idea and overview.

- Applications *(desktop, web, or mobile)* similar to the one you're developing, and what are the functionalities/features, and how they work *(if that information is available)*.

- A Literature Review of Academic publications *(papers/books/articles)* relevant to the problem you're trying to solve and the approach you're trying to implement *(at least 5 resources)*. You may find them by searching using Google Scholar.

Proposed Solution & Dataset:

- Main functionalities/features *(from the users' perspective)* in your proposed software/solution *(can be explained using a use-case diagram)*.

- If applicable to your project, the Dataset employed *(preferably a publicly available dataset)*.

30303060104694

Applied Algorithms:

- All the details of the AI/Machine-Learning algorithm(s)/approach(es) used to develop your project *(can be explained using block diagrams)*.

Experiments & Results:

- The Experiments, testing, and the results *(including plots of the evolution or training if applicable)* and samples of the output *(and how did you test the solution)*.

Analysis, Discussion, and Future Work:

- Analysis of the results, what are the insights?

- What are the advantages / disadvantages?

- Why did the algorithm behave in such a way? What might be the future modifications you'd like to try when solving this problem?

303030601046 94