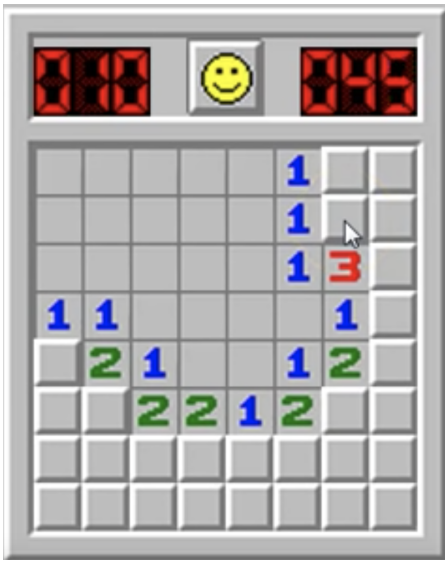

Minesweeper

Spring 2021, CS171

Created By : Sakshi Agarwal (sakshia1@uci.edu), Mike Heddes (mheddes@uci.edu), Sadeem Al Sudais (salsudai@uci.edu)

Before you begin implementing your AI:

Once you have followed the steps to set up the environment and you have access to the codes on your system/gitlab, we recommend you to thoroughly go over the source codes - WorldGenerator.py (generates different worlds for the game. You should read this to get an idea of how these worlds are generated.), individual files in the Minesweeper project folder for a particular language you are comfortable with. This will help you understand the interaction between the AI and the board, how the board works, what are the states of the board as you/your AI plays the game. We strongly recommend you to run the program in manual/random AI in debug mode so as to monitor the states of the board as you play the game. Once you feel confident about the code for Minesweeper and is no longer a black box for you, think about solving it!

If you read (skim over) the code provided carefully, a lot of your trivial questions will hopefully be answered and hopefully implementation will become easier!

Reminder - Student Questions

If you have any questions, first check the documents provided. If there is no answer in this document, you should go to the piazza and click project. Before you post any questions, you should first use search bar Search or add a post... to find existing questions related to your question. If there are no answers, then you do a new post. If the question is good and not in this document and piazza before, your question will be marked as good and receive credit.

Introduction

In this programming assignment, you will be tasked with implementing a Minesweeper AI Agent, which should be able to play and solve the Minesweeper game. You will have the choice of programming in C++, Java, or Python.

The only thing you need to write is your AI. We will provide other things for you.

Game Mechanics

Wikipedia:https://en.wikipedia.org/wiki/Minesweeper_(video_game)

Here is a good video to learn how to play. We recommend you to go through it if you don't know how to play the game.

Youtube:https://www.youtube.com/watch?v=7B85WbEiYf4

However, there are a few differences designed to better evaluate your agent which will become apparent as you play with and familiarize yourself with the game. In Minesweeper, you are given a board that is set up as a 2D grid of tiles. Each tile covers either: (1) a hint number (that tells how many mines are around that tile) or (2) a mine. Ultimately, your agent's goal is to uncover all tiles which do not contain a mine. A more concrete definition of the game is given by the following PEAS description.

An additional note : The first tile for you will be uncovered with a '0' indicating that the nearby tiles don't have any mines. This is to prevent your AI from failing on the first move.

Performance Measure

- The performance measure of your agent will be a score calculated based on the number of worlds your agent has completed. Points are awarded to your agent only if it successfully solves the entire world. Each difficulty level has a different weight.
- The game ends when your agent chooses to leave the game or if your agent uncovers a mine. In either of these cases you'll get a zero.

Environment

Refer to the WorldGenerator folder for any further clarification/implementation

- Each difficulty level has a different dimension and number of mines (generateTournament.sh generates these boards):
    - Beginner: 8 row x 8 column with 10 mines

    - Intermediate: 16x16 with 40 mines
    - Expert: 16x30 with 99 mines
- The board begins with 1 random tile already uncovered and presumably safe. (read WorldGenerator.py for more information)
- Mines are randomly placed throughout the board.
- Your agent dies when it uncovers a mine.

Actuators

Refer to the file Agent for any further clarification/implementation

- Your agent has 4 moves:
    - (1) The action UNCOVER reveals a covered tile.
    - (2) The action FLAG places a flag on a tile.
    - (3) The action UNFLAG removes a flag from a tile if that tile has a flag.
    - (4) The action LEAVE ends the game immediately.
- The actions UNCOVER, FLAG, and UNFLAG are to be coupled with a pair of coordinates which allows the agent to act on a single tile.

Sensors

- Your agent will receive only one percept:
    - Following an UNCOVER action, your argent will perceive the hint number associated with the previous UNCOVER action. This number represents how many mines are within that tile's immediate neighbors.
    - Following a FLAG or UNFLAG action, your agent will perceive -1.

Understanding the Tournament

● After you submit your Final submission and the deadline passes, your agent will be entered into a tournament with your classmates.

● The tournament checks to make sure you followed all the instructions correctly, then runs your agent across 3N (say 10) random worlds with three different difficulty levels of N worlds each. Every agent is run on the same 3N worlds to ensure fairness.

● Your agent's total score is calculated and a scoreboard is constructed that will be made available.

● Your agent will be timed-out if it hangs for longer than 5 mins per world.

● After the scoreboard is constructed, scores are checked for any illegal submissions. These include two agents with the same score.

● Late submissions won't be entered into the Tournament, and therefore, receive no bonus points.

Scoring Explanation

The scoring (20% of course grade) will depend on:

1) Team Formation - 4%

2) Submissions: Minimal, Draft, Final - 12% (4% each)

3) Final Report - 4%

4) Tournament Bonus - 2%

Late submissions :

You are given 1 late day total per quarter. You have 5 deadlines. You can be late (up to 24 hours) on a single deadline. So, e.g. let's say you were on time for deadline 1, then 1 day late for deadline 2, then after that if you are late, it is automatic 0 for that submission.

1. Team Formation

Due: 04/25/2021 11:59pm

You can either form a team of 1 or 2. But you need to submit your team formation through Canvas. The link to upload your team formations will be uploaded shortly.

The submission of Team Formation must follow strict rules:

- Please use only letters,digits in your team name
- Team names may not contain any personally identifiable information, especially not the students' name or ID nor any part thereof. Doing so would result in a violation of legally protected student private confidential information, because we will release the results of the Tournament publicly by team name.
- Please Don't Use Spaces,"_"(underscore),or any other special characters

- Please follow this format or points will be deducted:

TeamName: *TeamName*
Member1: *FirstName LastName*
Member2: *FirstName LastName*

- Example:

TeamName: ExampleAI
Member1: John Doe
Member2: Jane Doe

- If done without a partner, please delete the Member2 line

You will lose points (up to 100%) if you don't follow the stated rules above.

2. Submissions:

Minimal

Due:05/02/2021 11:59pm

Complete 20% out of N (say 10) Easy worlds (5x5 with 1 mine).

Draft

Due:05/23/2021 11:59pm

Complete 30% out of N Beginner worlds (8x8 with 10 mines) and 15% out of N Intermediate worlds (16x16 with 40 mines).

Final

Due:06/06/2021 11:59pm

Complete 60% out of N Beginner worlds (8x8 with 10 mines), 50% out of N Intermediate worlds (16x16 with 40 mines), and 10% out of N Expert worlds (16x30 with 99 mines)

Some Important grading policy to note:

- You will get full credit if you meet these requirements per submission.

- You will lose points if you don't meet the requirement based on the fractional weight of your completed worlds against the required completion.

- Each difficulty requirement has the same weight that sums up to 100%, for example: in Minimal, Easy has a weight of 100%. In Draft, Beginner and Intermediate has a weight of 50% each. In Final, each difficulty has a weight of approximately 33.3%

- For example, for Final, if your AI completes >= 60% Beginner worlds, you will get 100% out of 33.3% (Beginner weight). Else, you get (completed/required) x 100% out of 33.3%. This applies to all three difficulties.

Please note that your code should run in openlab. Make sure you test this before your submission. This seldom happens but it does: it runs in your computer environment, but it doesn't run in openlab (and openlab is where the grading will take place), so please test it for consistency. No points will be awarded if your code doesn't run on openlab.

3. Final Report

Due:06/10/2021 11:59pm

If you write each section in clear, logical, technical prose, you will get 100% credit. The template for the Final Report is given at the bottom of this document. Please strictly follow the template and fill in the details that are asked.

Points will be deducted if -

1. **The 'brief' description you provide is more than 1 page. Please note that this report only asks of a description of your approach and nothing else. Try to organize your answer and write it to the point.**
2. **If your report has any screenshots, irrelevant background information, or copy-pasted information from online resources like wikipedia etc.**

We require only **the description of your approach** in the report.

4. Tournament Bonus

The tournament score percentage is calculated as follow:

- The maximum completion score in class for each difficulty will be set as the bar, which is 100%. Therefore, the bars can originate from different teams.
- An example bar would be: 99% Beginner worlds completed (by Team A), 85% Intermediate worlds completed (by Team X), 40% Expert worlds completed (by Team Z).
- The weight for each difficulty will follow the Final submission explained above.
- Sample calculation: Team B achieves 66% Beginner completion, 68% Intermediate completion, and 20% Expert completion. Thus, Team B's tournament score percentage will be (66/99) x 33.3% + (68/85) x 33.3% + (20/40) x 33.3%, which is 65.49%.
- This percentage is not an absolute score and will only be used to rank your AI from others.

Your AI's percentage score will be compared to other students in a ranking system. The tournament bonus is between 1-10, where the top 10% highest scoring teams will get a 10, the second 10% will get a 9, and so on. **Generally, a 10 means the top 10% of the students get 10% of total project pts, i.e. that would be 10% of 20 = 2. Next 10% get 9% of 20 = 1.8, next 10% get 8% of 20 = 1.6, etc..**

Common Mistakes

Below are the common mistakes that students often make when submitting.

- Leaving 'print' or 'cout' statements in the source codes. Please delete these because it will create problems for the grader in both time and script.
- Giving additional information or text in the Team Formation submission. Please only fill the required info.
- Not testing in openlab. This seldom happens but it does: it runs in your computer environment, but it doesn't run in openlab (and openlab is where the grading will take place), so please test it for consistency.
- Only submitting the source code(s)/ Not submitting a zip file. Please use the make/ make all command provided, as it will make a zip file with your team name (<team_name>.zip).

Minesweeper Final AI Report

Team name_____
Member #1 (name/id)_____
Member #2 (name/id)_____

I. Minimal AI

I.A. Briefly describe your Minimal AI algorithm. What did you do that was fun, clever, or creative?

I.B Describe your Minimal AI algorithm's performance:

Board Size	Sample Size	Score	Worlds Complete
5x5			
8x8			
16x16			
16x30			
Total Summary			

II. Final AI

II.A. Briefly describe your Final AI algorithm, focusing mainly on the changes since Minimal AI:

II.B Describe your Final AI algorithm's performance:

Board Size	Sample Size	Score	Worlds Complete
5x5			
8x8			
16x16			
16x30			
Total Summary			

III. In about 1/4 page of text or less, provide suggestions for improving this project (*this section does <u>NOT</u> count as past of your two-page total limit.*)