OBLIVION

Artificial Intelligence Project Report

k22-4543	Syeda Fizza
k22-4219	Zehra Waqar
k22-4237	Zahab Jahangir

Submitted to: Sir Shafique Rehman
Submission Date: 11, May 2025

Executive Summary

- **Project Overview:**
  This project aimed to create an AI assisted mystery game using clues and probabilistic reasoning to solve the case. The system made use of Bayesian probability to dynamically update the likelihood of the suspect's truthfulness based on new evidence, and Reinforcement Learning (RL) to refine the system's ability to detect lies over time. Rule-based logic, such as "the killer always lies," is also incorporated to guide suspect behavior.

Introduction

- **Background:**
  Our game was inspired by a board game *Clue* in which 2-8 players can play at a time. Our game is a single player with game's logic of generating random scenarios with random suspects as killers and random weapon logic as in the original game, a card for the killer from a deck of suspects and a weapon from the deck of weapons is selected and kept aside, which the players are to guess. However the rest of the game logic is novel by placing greater lie probability for the killer who always lies and other clues based on motives, height, hair color and weapon are provided for the player.
- **Objectives of the project**
  The goal was to make the robot assistant pip guess the right killer using bayes theorem and csp and use reinforcement learning to improve lie detection accuracy.

Game Description

- **Original game rules:**
  The original game is a multiplayer board game which can be played by 2-6 players. There are cards 6 for suspects, 6 for weapons and 9 for rooms. At the start of the game one card is taken out from each category and kept in an envelope unknown by the players. The rest of the cards are shuffled and divided equally among players. The

players are also given a list where they can keep track of the cards they know cannot be the ones in the envelope. The players can make as many rooms as they wish in the room they suspect the murder happened. But can only make one accusation. The first one to make the correct guess wins. If the accusation made is wrong tho the player who made the accusation loses.

- **Innovations and modifications**
  Oblivion, although inspired from *Clue*, is very different. It is a one player game in which a randomly selected suspect is the killer and a weapon too is randomly selected before the game begins, unlike the original game there is no room in our version. The player in the first stage gets the inspector's findings, introducing all the potential weapons. In stage 2, the player interrogates the suspects to find the potential motives and their alibis each may have. Stage 3, is where the 4 minute timer starts. In this stage suspect's tell their theories about where they believe other suspects were at the time of murder. Stage 4 is testing the alibis and checking who lied and who told the truth. *Pip* the AI assistant calculates on the basis of the lie probabilities who is most likely to be lying and hence who's alibi should be verified. In the last stage the player is given a hint about the possible weapon and *Pip* suggests 2 potential suspects calculated through on the basis of all the probabilities. The player has 3 guesses to guess the killer and weapon. If guessed correctly the player wins else they lose. Based on time taken and tries made, the player is given a score and the winner is added to the leaderboard.

AI approach and methodology

- **AI Techniques Used:**
  1. **Bayesian Network:** update suspect honesty based on clues.
  2. **Reinforcement learning:** improve lie detection of suspects using Q-learning to optimize success rate over time.
  3. **Constraint satisfaction problem:** logically deduce which suspects are consistent with the constraints of the crime,

helping the system make intelligent decisions and give smarter clues.

4. **Rule based:** simulate expert-system reasoning.
5. **Search space reduction:** AI-style pruning of irrelevant suspects.

- **Algorithm and Heuristic Design:**
  1. **Bayesian Network:** makes use of clue discoveries to recalculate and update probabilities.
  2. **Reinforcement Learning (RL):** makes use of probability to recommend which alibi to check (+0.2 if correct -0.2 if wrong) .
  3. **Rule-based constraints:** There are 2 rules. One is the hard rule that is "Killer always lies" and the other is heuristics to prioritize suspects with both motive and opportunity.
- **AI Performance Evaluation:**

  The accuracy of AI predicting the Top-1 correct was 95.60%, while Top-2 correct was 4.40%.

Game Mechanics and Rules

- **Modified game rules:**
  1. **Number of players:** original game is multiplayer but oblivion is a single player game.
  2. **Time limit and number of guess:** There is no time limit in original game and only one final accusation whereas in our game the player has to guess within 4 minutes and has 3 tries.
  3. **Deduction style:** In *Clue*, players deduce the solution by eliminating suspects and weapons that cannot be in the envelope. In *Oblivion*, players are given positive clues that help them identify the most likely suspect and weapon based on traits and patterns, following a forward reasoning approach.
- **Turn-based Mechanics**

In oblivion players have 3 turns to guess the correct killer and weapon. If a player fails to guess within the 3 turns or within the time limit they lose.

- **Winning Conditions:**

  Player wins if they guess the correct killer and weapon within the time limit and within the 3 tries. Their score is calculated based on the number of guesses they made before the right one and the amount of time they used and they are added to the leaderboard if they provide their name after winning.

Implementation and Development

- **Development Process**

  The game logic was written in Python making use of the random library to generate random scenarios (killer and weapon) in each game. The uuid library was also used to generate a unique identifier for each player session. Lastly flask was used to connect the backend with the react frontend.

- **Programming Languages and Tools:**
  - **Language used:** python, javascript
  - **Libraries:** random, flask, uuid, pymongo, re
  - **Tools:** React, Vite
  - **Database:** MongoDB
- **Challenges Encountered:**

  The first challenge we faced was determining the working of the game. We were originally planning a chat bot to assist the players with predefined questions leading to the suspect. However our original model seemed to lack the AI implementation as it would have been solely rule based so we had to figure out another way to implement.

Another challenge we encountered was with the db as the original database that had been created was not accessible by the other members and hence another database had to be created to provide access to all to be able to test whether it worked correctly after integration with the rest of the code.

Team Contributions

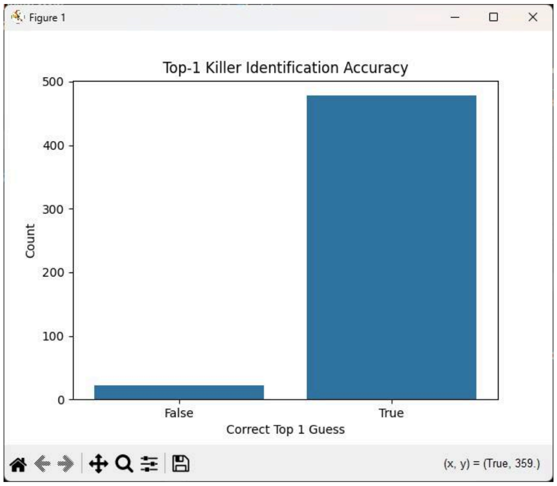
Team Member	Contribution
Zahab Jahangir	Developed the entire backend, including AI logic, Flask integration, and database management.
Zehra Waqar	Handled complete frontend development, ensuring an interactive and engaging user experience.
Syeda Fizza	Compiled the report, contributed to the game's structure and flow as a Game Flow Designer, and conducted performance testing

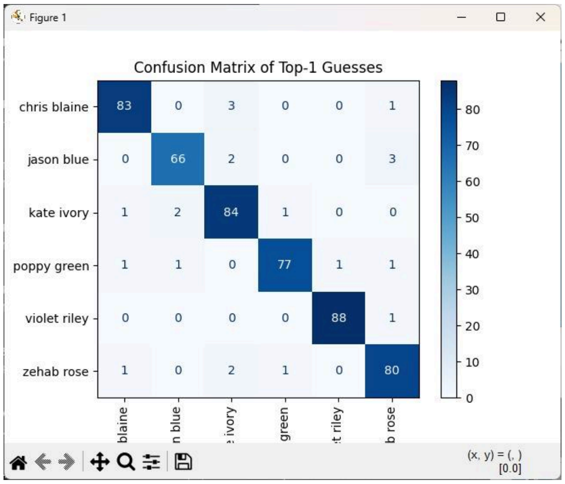
Results and Discussion

- **AI Performance:**

The AI system generates suggestions for the top two most likely suspects in each case. We conducted 500 test runs, and the results indicate that the accuracy of the AI in correctly identifying the Top-1 suspect was 95.60%, while the Top-2 suspect accuracy was 4.40%, as illustrated in Figures 1.a and 2.a.
Figure 1.b presents the confusion matrix for Top-1 accuracy, where the AI consistently identified the correct suspect in most instances. On the other hand, Figure 2.b shows the confusion matrix for Top-2

accuracy, where the AI's performance in selecting the correct suspects was significantly lower, as reflected by the reduced frequency of correct guesses.

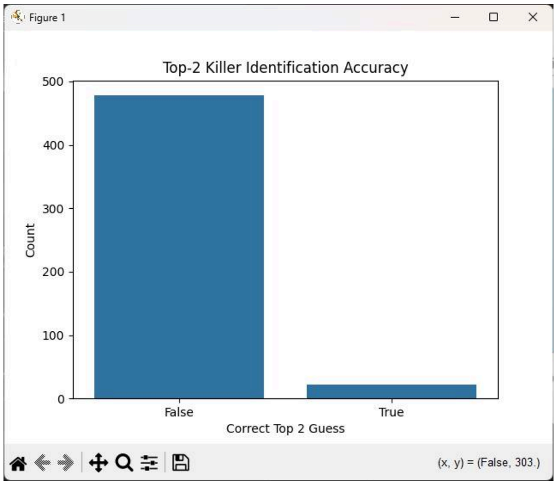


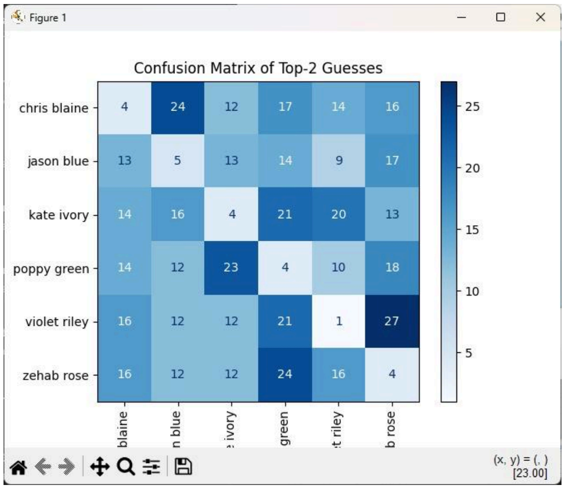


	blaine	n blue	ivory	green	t riley	b rose
chris blaine	83	0	3	0	0	1
jason blue	0	66	2	0	0	3
kate ivory	1	2	84	1	0	0
poppy green	1	1	0	77	1	1
violet riley	0	0	0	0	88	1
zehab rose	1	0	2	1	0	80

Figure 1.a (Killer identification accuracy of Top-1 guess)

Figure 1.b (Confusion matrix of Top-1 guess)





	blaine	n blue	ivory	green	t riley	b rose
chris blaine	4	24	12	17	14	16
jason blue	13	5	13	14	9	17
kate ivory	14	16	4	21	20	13
poppy green	14	12	23	4	10	18
violet riley	16	12	12	21	1	27
zehab rose	16	12	12	24	16	4

Figure 2.a (Killer identification accuracy of Top-2 guess

Figure 2.b (Confusion matrix of Top-2 guess)

References

- Flask: https://flask.palletsprojects.com
- React: https://reactjs.org
- Clue game rules: https://www.hasbro.com/common/instruct/clueins.pdf