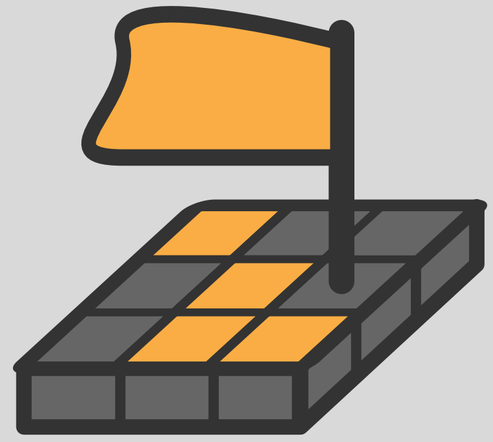
CS 246 - FALL 2023
ARTIFICIAL INTELLIGENCE (SEC. A & B)

LET`S PLAY MINESWEEPER

START

VRAM PAPYAN
LILI KOSTANYAN
MERI ASATRYAN
JENI BALABAN



ABOUT THE GAME...

THE ORIGINAL GAME CALLED **MINE** WAS CREATED BETWEEN 1989 AND 1990 BY ROBERT DONNER, AND HE USED CURT JOHNSON'S CODE .

HOWEVER, MINESWEEPER GAINED A LOT OF INSPIRATION FROM A "LESSER KNOWN, TIGHTLY DESIGNED GAME", **MINED-OUT** BY IAN ANDREW IN 1983.

RULES!

GAME START: THE PLAYER SEES A BOARD FILLED WITH SQUARES AT THE BEGINNING OF THE GAME.

INTERACTION: THE PLAYER CAN INTERACT WITH THE BOARD BY CLICKING ON THESE SQUARES.

PROBING ACTION: THIS CLICKING ACTION IS KNOWN AS PROBING.

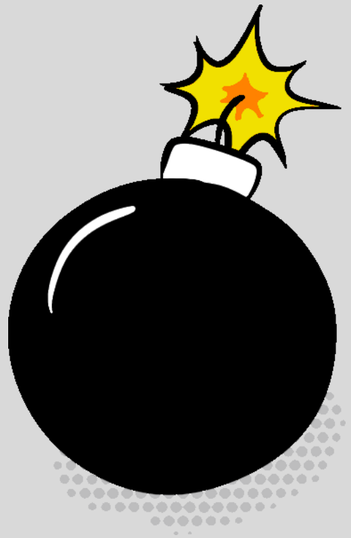
HIDDEN NUMBERS: EACH SQUARE HIDES A NUMBER THAT IS REVEALED ONLY AFTER PROBING.

NUMBERS MEANING: THE REVEALED NUMBER INDICATES THE COUNT OF BOMBS IN THE ADJACENT SQUARES.

HIDDEN MINES: SOME SQUARES ALSO CONCEAL A MINE, WHICH IS REVEALED UPON PROBING.

KNOWN ALGORITHMS

- NAIVE SINGLE POINT
- CELLULAR AUTOMATON
- BASIC LIMITED SEARCH ALGORITHM
- THE DOUBLE SET SINGLE POINT (DSSP)
- IMITED SEARCH WITH PROBABILITY ESTIMATES
- MINESWEEPER CONSISTENCY PROBLEM



OUR IMPLEMENTATIONS

- CONSTRAINT SATISFACTION PROBLEM
- STOCHASTIC (PROBABILITY BASED MOVE)
- KNOWLEDGE BASE

MEET OUR AGENT- OPPIE

FIRSTLY, WE DECIDED TO FORMULATE THE GAME AS A CSP

VARIABLES: THE CELLS OF THE MINESWEEPER GRID ARE THE VARIABLES, EACH WITH ITS OWN STATE (LIKE 'HIDDEN' OR 'REVEALED') AND VALUE ('MINE' OR 'SAFE').

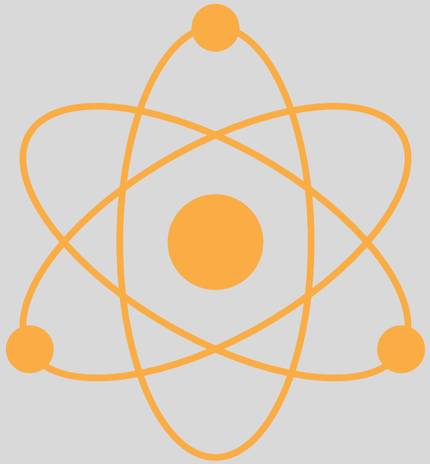
DOMAINS: THE DOMAINS ARE THE POSSIBLE STATES OR VALUES EACH CELL CAN HAVE - EITHER HIDDEN OR REVEALED, AND EITHER CONTAINING A MINE OR BEING SAFE.

CONSTRAINTS: THE GAME HAS RULES (CONSTRAINTS) THAT LIMIT HOW CELLS CAN BE REVEALED. FOR EXAMPLE, A CELL'S NUMBER INDICATES THE COUNT OF MINES IN ADJACENT CELLS, AND CELLS NEXT TO MINES ARE APPROACHED CAUTIOUSLY.

OVERVIEW

OUR AGENT ENHANCES THE CLASSIC MINESWEEPER GAME WITH AN **ARTIFICIAL INTELLIGENCE ASSISTANT.** THE AI USES **PROPOSITIONAL LOGIC** AND **INFERENCE RULES** TO DETERMINE SAFE CELLS DURING GAMEPLAY, GAINING MORE KNOWLEDGE WITH EACH MOVE.

FOR INSTANCE, IT CAN IDENTIFY ALL NEIGHBORING CELLS AS **SAFE** IF A CELL HAS **ZERO** SURROUNDING MINES, OR MARK A CELL AS A MINE IF IT'S THE ONLY UNREVEALED CELL NEXT TO A CELL WITH 1 MINE.



CONT.

CONSTRAINT PROPAGATION:
THE MINESWEEPER AI UTILIZES CONSTRAINT PROPAGATION TO MAKE SAFE MOVES BASED ON THE REVEALED CELLS AND THE NUMBER OF NEARBY MINES.

THE MINESWEEPERAI CLASS INCLUDES A MAKE_SAFE_MOVE() METHOD, WHICH SELECTS A SAFE MOVE BASED ON KNOWN SAFE CELLS AND CONSTRAINT PROPAGATION.

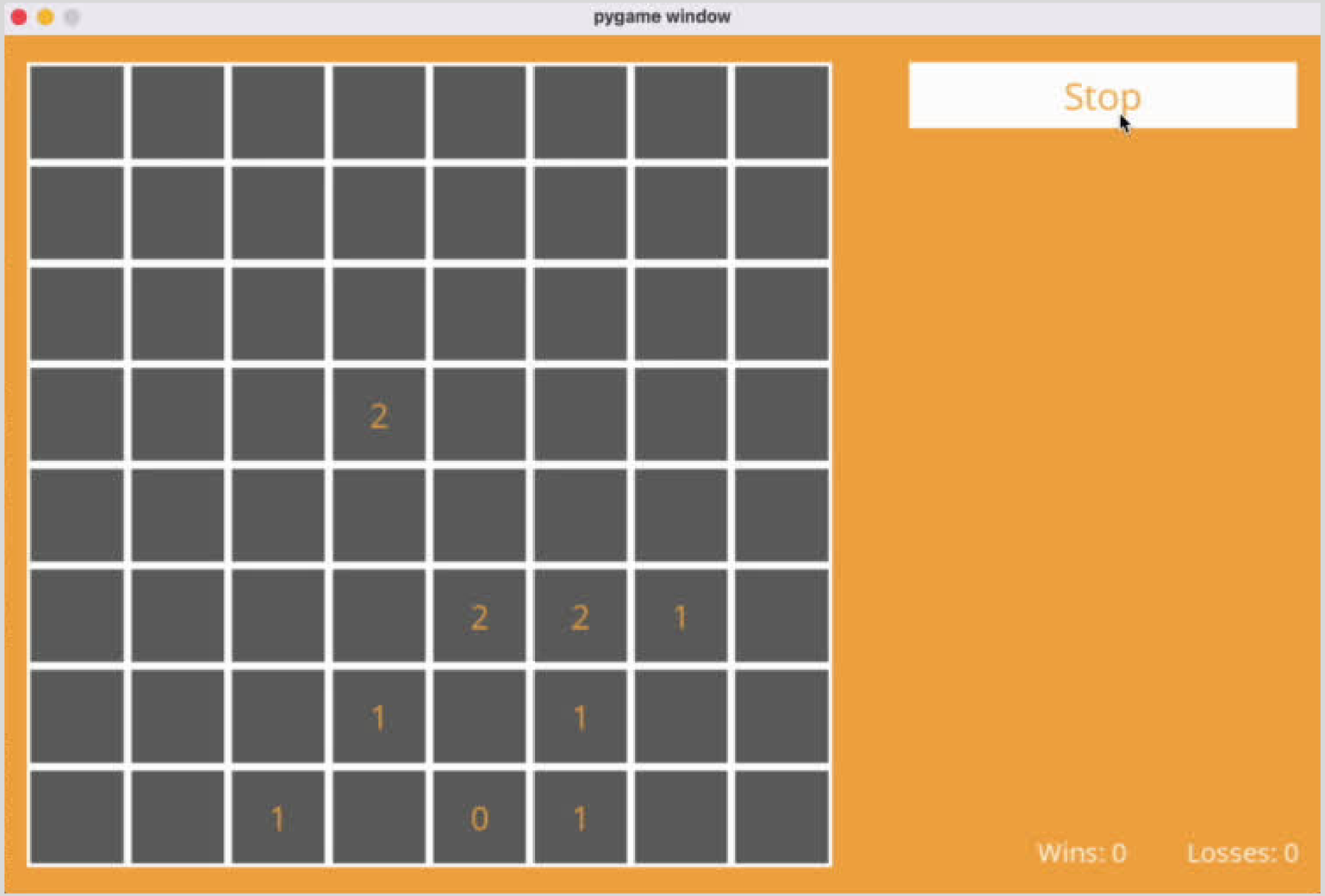
Key Features

AI ASSISTANCE: UNSURE ABOUT YOUR NEXT MOVE? LET THE AI DECIDE FOR YOU, THOUGH IT MAY OCCASIONALLY HIT A MINE DUE TO LIMITED INFORMATION.

AUTOPLAY MODE: WATCH THE AI PLAY THE GAME, STARTING WITH RANDOM MOVES AND GRADUALLY MAKING SAFER CHOICES BASED ON ACCUMULATED KNOWLEDGE.

INFERENCE VISUALIZATION: THE AI'S DECISIONS ARE HIGHLIGHTED FOR EASY UNDERSTANDING, WITH SAFE CELLS COLORED IN CYAN AND MINES IN RED.

A Quick Glance...



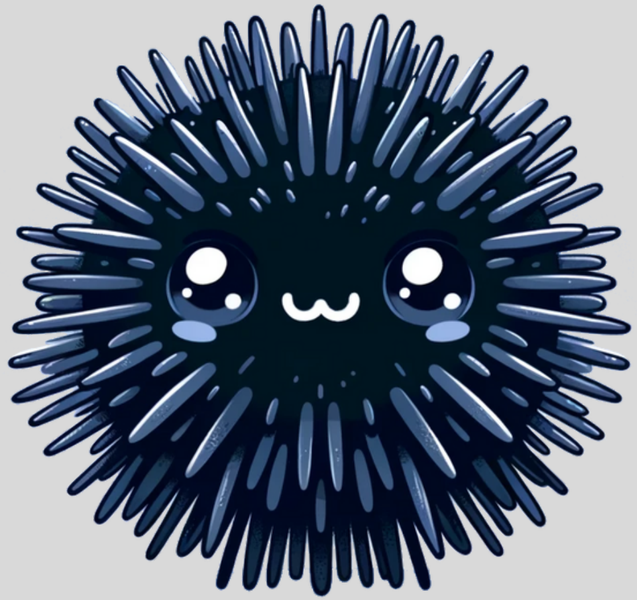
MEET NEXT AGENT- URCHIE

SECONDLY, WE DECIDED TO FORMULATE THE GAME AS A STOCASTIC

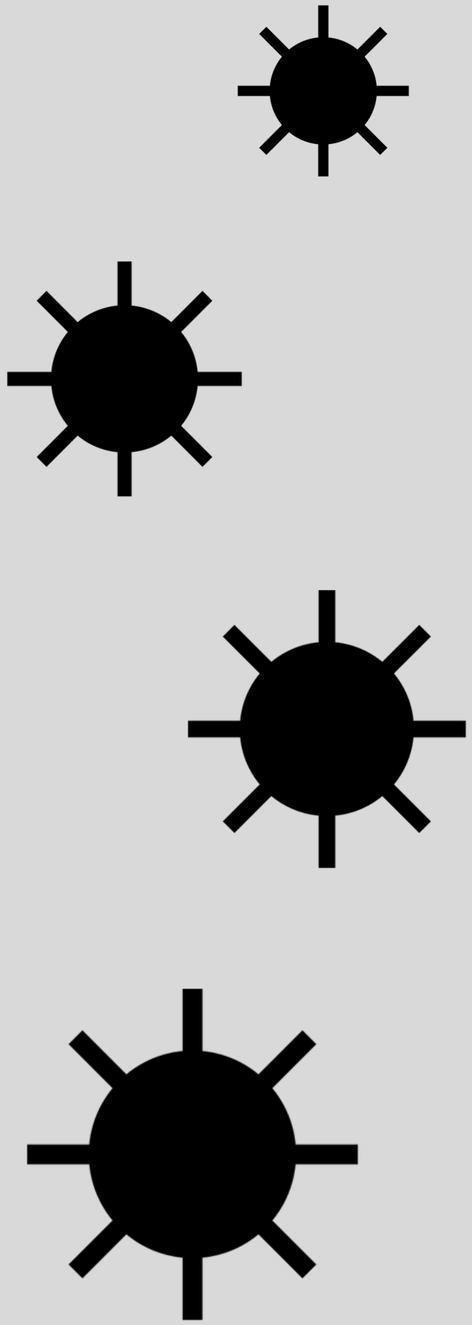
1. INITIALIZATION: THE ALGORITHM STARTS BY IDENTIFYING ALL POSSIBLE MOVES ON THE MINESWEEPER BOARD.

2. CALCULATING PROBABILITIES: FOR EACH AVAILABLE CELL, THE ALGORITHM CALCULATES THE PROBABILITY OF IT BEING A MINE BASED ON ITS NEIGHBORS. THIS IS DONE IN THE CALCULATE_MINE_PROBABILITY( ) METHOD.

THE PROBABILITY IS DETERMINED BY CONSIDERING THE NUMBER OF SAFE NEIGHBORS AND THE REMAINING NUMBER OF MINES TO BE FOUND.



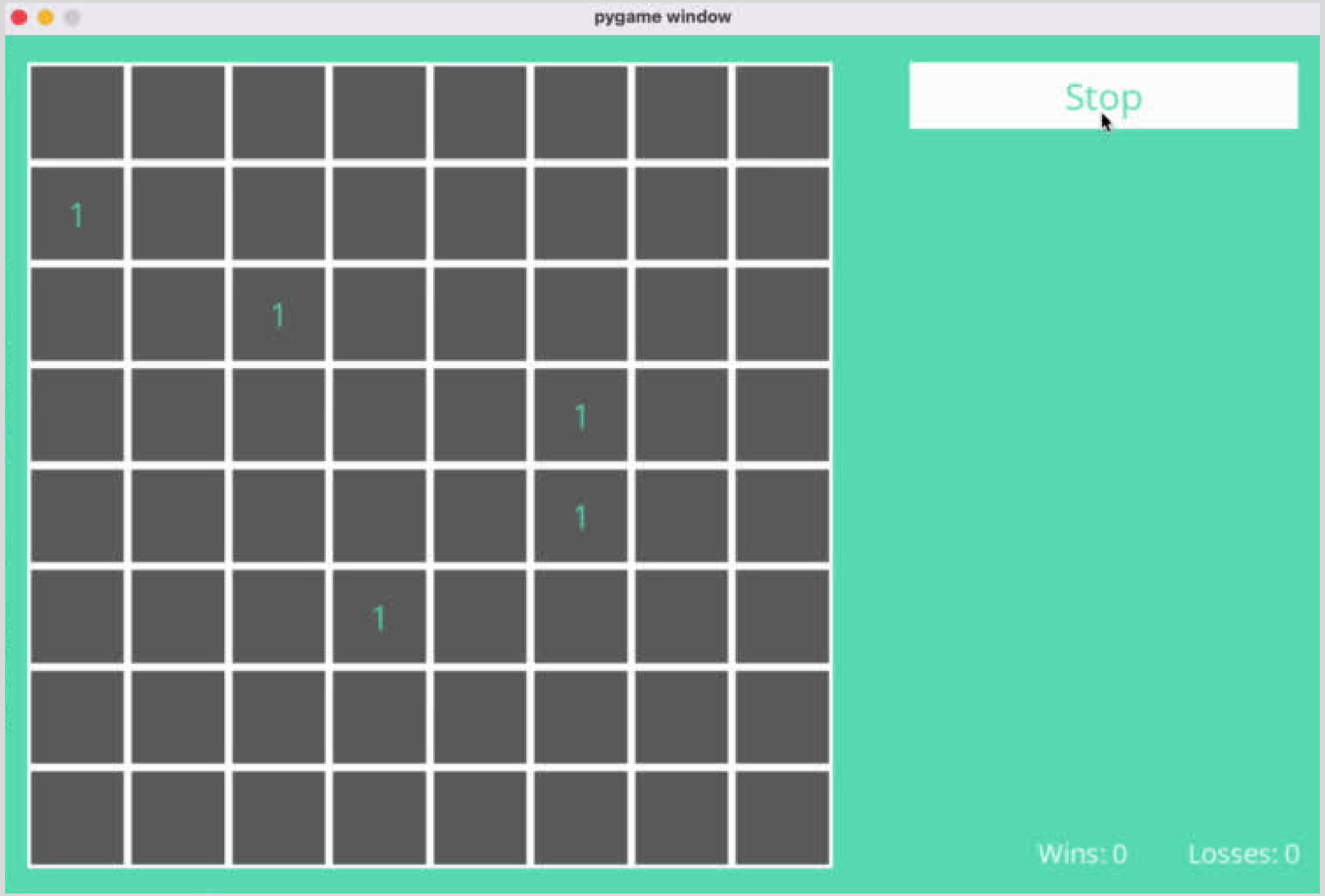
CONT.



3. SORTING BY PROBABILITY: THE AVAILABLE CELLS ARE SORTED IN ASCENDING ORDER OF THEIR PROBABILITIES, WITH THE LEAST LIKELY CELLS TO BE MINES COMING FIRST.

4. SELECTING THE MOVE: THE ALGORITHM CHOOSES THE CELL WITH THE LOWEST PROBABILITY AS THE NEXT MOVE. THIS DECISION-MAKING PROCESS IS IMPLEMENTED IN THE PROBABILITY_BASED_MOVE( ) METHOD.

A Quick Glance...



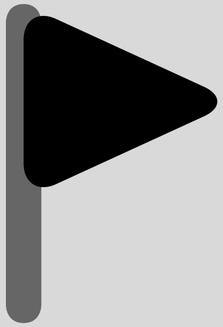
KNOWLEDGE BASE

KNOWLEDGE BASE MAINTENANCE: THE ALGORITHM MAINTAINS A KNOWLEDGE BASE WITH LOGICAL STATEMENTS ABOUT CELL POSITIONS AND MINE COUNTS IN MINESWEEPER.

I**NFORMATION UPDATING:** IT UPDATES THIS KNOWLEDGE BASE WITH NEW INFORMATION OBTAINED DURING GAMEPLAY, IDENTIFYING SAFE AND MINE-CONTAINING CELLS.

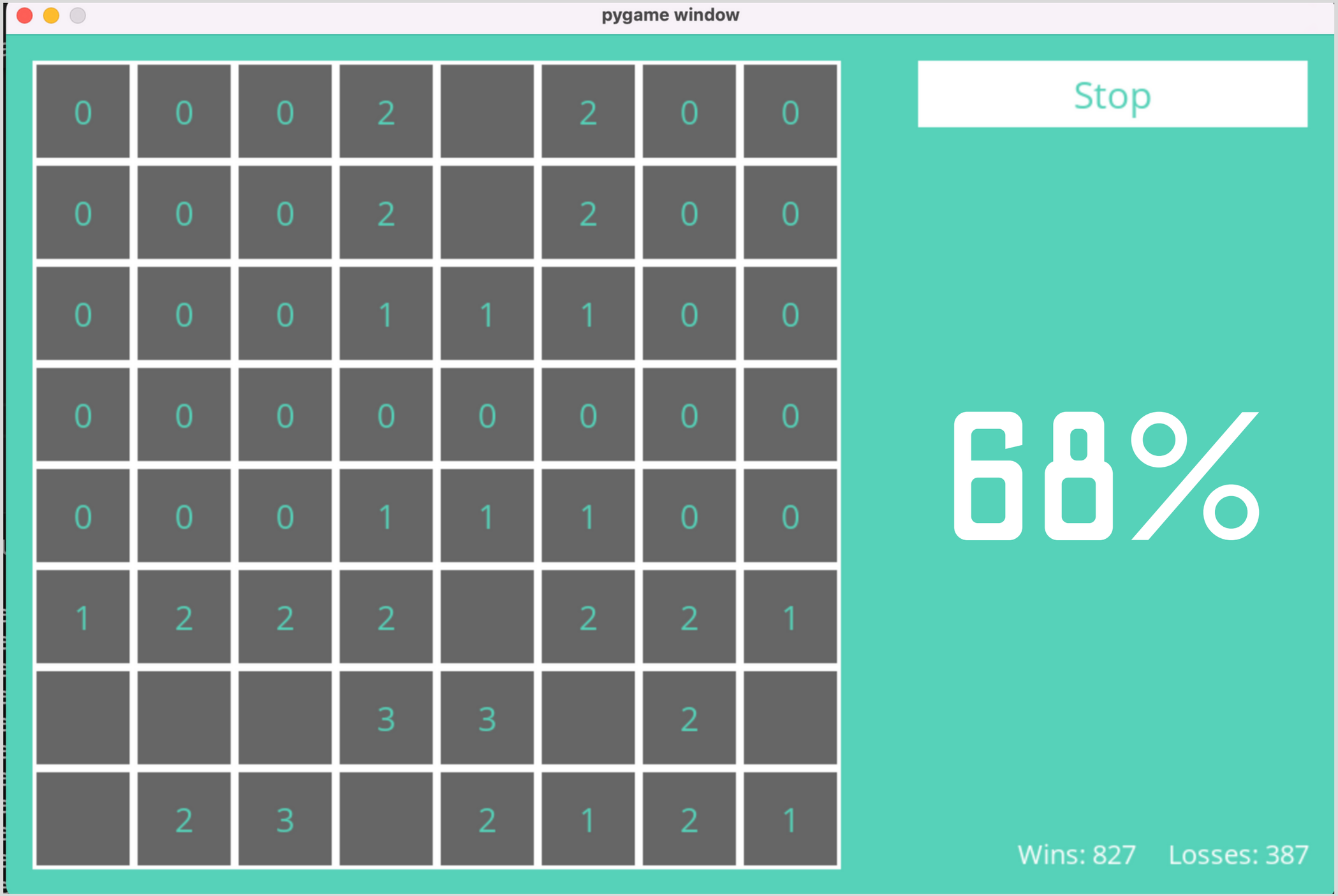
INFERENCE AND CONCLUSION: THE ALGORITHM ANALYZES AND COMPARES SENTENCES IN ITS KNOWLEDGE BASE TO MAKE INFERENCES ABOUT THE GAME BOARD.

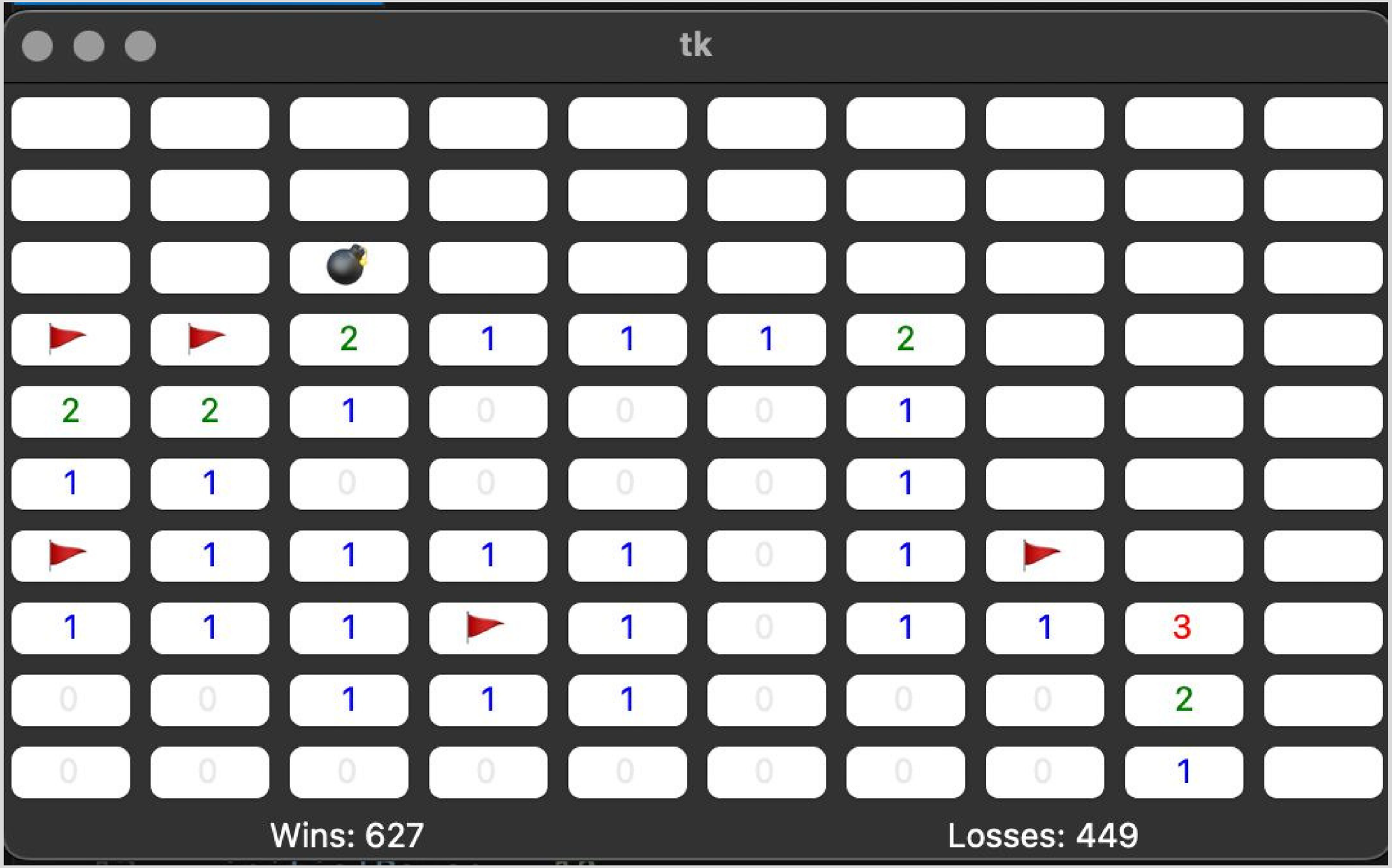
CONT.



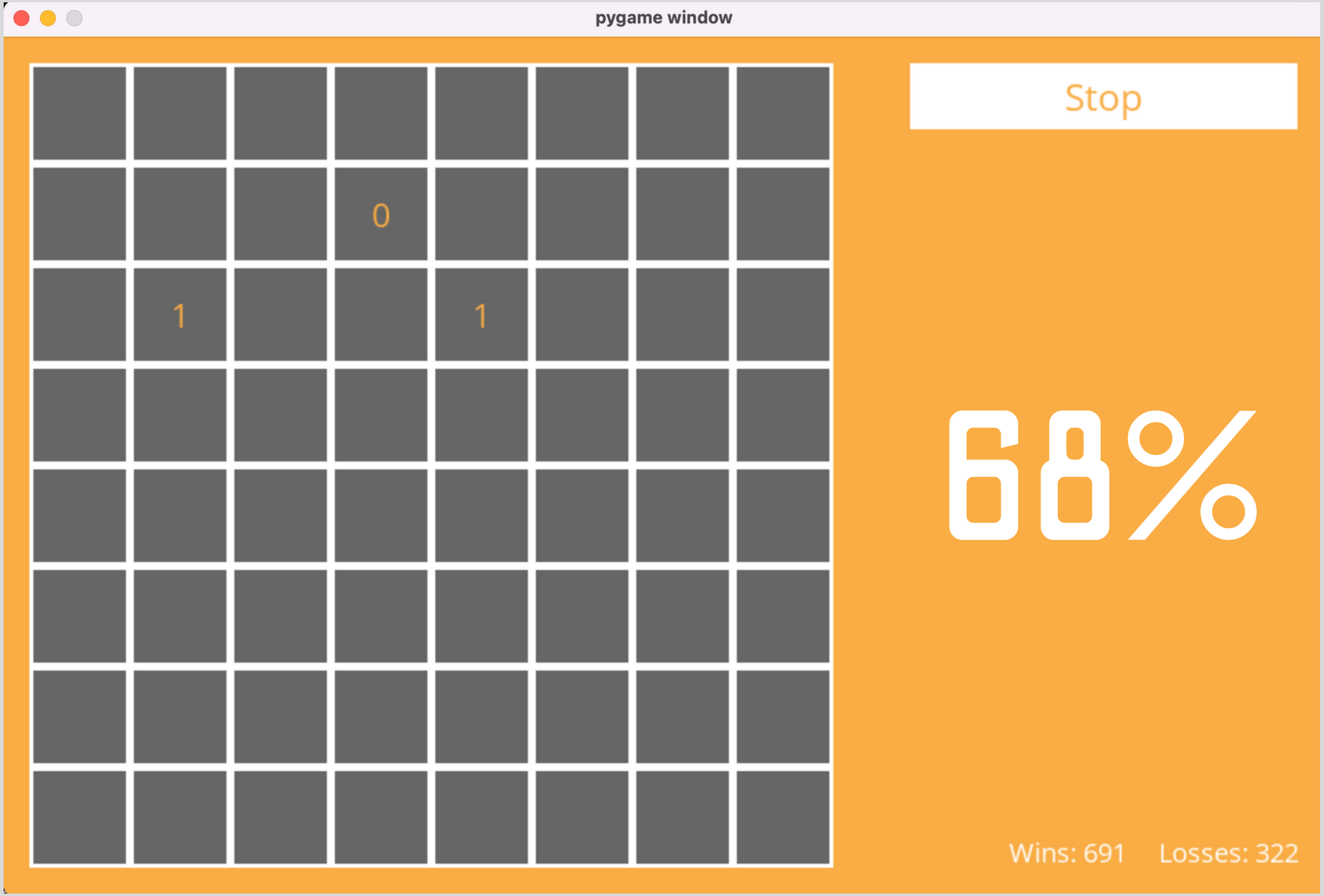
DECISION MAKING: MOVES ARE DECIDED USING A PROBABILITY-BASED APPROACH, WITH RANDOM MOVES AS A FALLBACK WHEN CLEAR CHOICES ARE NOT EVIDENT.

OVERALL STRATEGY: THE ALGORITHM USES A COMBINATION OF LOGICAL REASONING AND PROBABILISTIC ASSESSMENT TO STRATEGICALLY NAVIGATE THE MINESWEEPER BOARD, AIMING TO AVOID MINES.


pygame window
0 0 0 2 2 0 0
0 0 0 2 2 0 0
0 0 0 1 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0
1 2 2 2 2 2 1
3 3 2
2 3 2 1 2 1
Stop
68%
Wins: 827 Losses: 387



58%


pygame window
0
1
1
Stop
68%
Wins: 691
Losses: 322

THANK YOU!