CONNECT FOUR

CMPE 260 - PROJECT PRESENTATION

Abhishek Bais, Haley feng, princy joy, Shannon phu

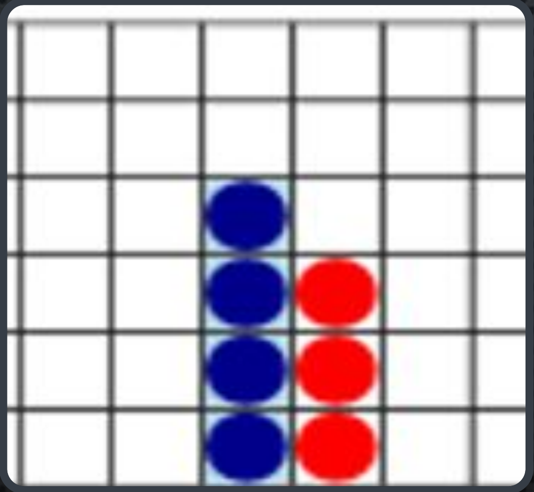
Graduate students software engineering, SJSU

OUTLINE

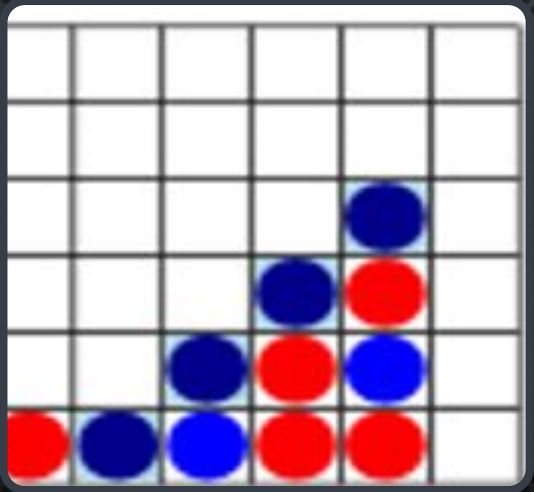
- The GAME
- Problem Statement
- Motivation
- Methodology
- Results
- Demo

THE GAME

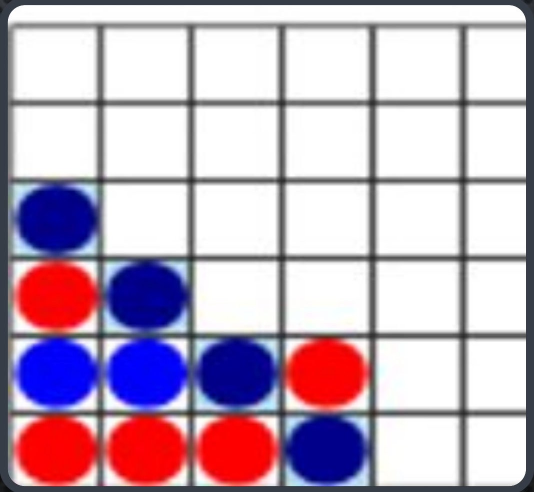
- Connect Four is a popular two player game
- Each player takes turns to drop a selected colored piece in a 6x7 grid
- First player to form a 4-in-row connection wins



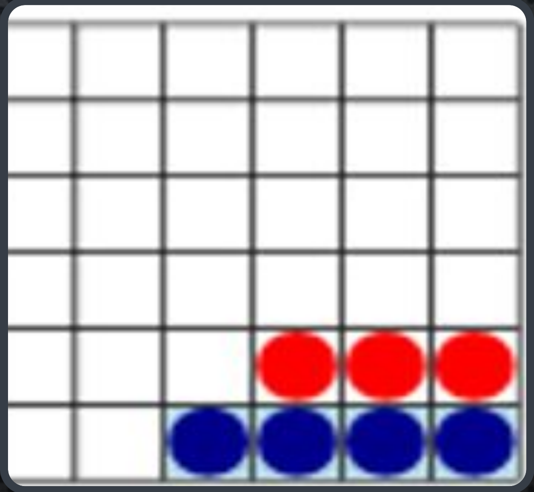
4-IN-ROW
VERTICAL



4-IN-ROW
DIAGONAL



4-IN-ROW
DIAGONAL



4-IN-ROW
HORIZONTAL

PROBLEM STATEMENT

CREATE A REAL-LIFE CONNECT FOUR GAMING EXPERIENCE FOR A HUMAN PLAYER AGAINST A COMPUTER AGENT

TRAIN REINFORCEMENT LEARNING (RL) GUIDED COMPUTER AGENTS VIA BATTLES AGAINST A COMPUTER AGENT THAT MAKES RANDOM MOVES AND AGAINST A COMPUTER AGENT THAT USES A NON-REINFORCEMENT LEARNING MINIMAX ALGORITHM TO PICK BEST COMPUTER AGENT IN TERMS OF WIN RATE AND EFFICIENCY

THE FOUR COMPUTER AGENTS CONTRASTED ARE

MINIMAX AGENT

USES A BACKTRACKING, RECURSIVE ALGORITHM USED IN GAME THEORY TO MAKE MOVES THAT RESULT IN MAXIMUM IMMEDIATE GAIN

MONTE CARLO AGENT

USES REINFORCEMENT LEARNING TO LEARN DIRECTLY FROM GAME EXPERIENCES WITHOUT USING ANY PRIOR MARKOV DECISION PROCESS KNOWLEDGE

Q LEARNING AGENT

USES A REINFORCEMENT LEARNING OFF-POLICY VALUE BASED SCHEME BASED ON THE BELLMAN'S EQUATION TO LEARN THE VALUE OF OPTIMAL POLICY REGARDLESS OF ACTION

SARSA LEARNING AGENT

USES A REINFORCEMENT LEARNING ON-POLICY VALUE BASED SCHEME TO LEARN THE VALUE OF THE OPTIMAL POLICY BASED ON ACTION DERIVED FROM CURRENT POLICY

MOTIVATION

MODEL AN INTERMEDIATE COMPLEXITY (6X7 BOARD GAME)

MOST EXISTING REINFORCEMENT LEARNING MODELS IN THE GAMING CONTEXT ARE FINE-TUNED TO TIC-TAC-TOE (A SIMPLISTIC 3X3 BOARD GAME WITH A SMALL STATE SPACE) OR ALPHA-GO (A PROGRAM THAT PLAYS GO, A 19X19 BOARD GAME AFTER STORING 30 MILLION POSITIONS)

PROVIDE INSIGHTS INTO HOW REINFORCEMENT LEARNING GUIDED COMPUTER AGENTS PERFORM IN BATTLES AGAINST A COMPUTER AGENT THAT MAKES RANDOM MOVES AND IN BATTLES AGAINST A COMPUTER AGENT THAT USE MINIMAX (A BACKTRACKING, RECURSIVE GAME THEORY ALGORITHM) IN TERMS OF WIN RATE AND EFFICIENCY

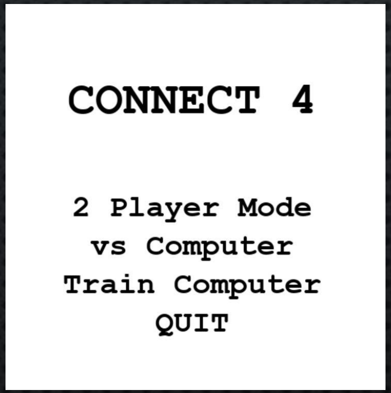
METHODOLOGY – GAME SETUP

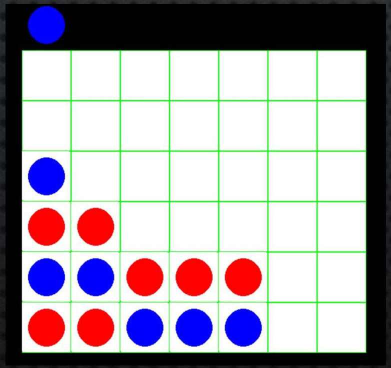
THE GAME WAS SETUP TO BE PLAYED IN THREE MODES

MODE 1 - Single Player Mode (Human Player vs Reinforcement Learning Guided Computer Agent)

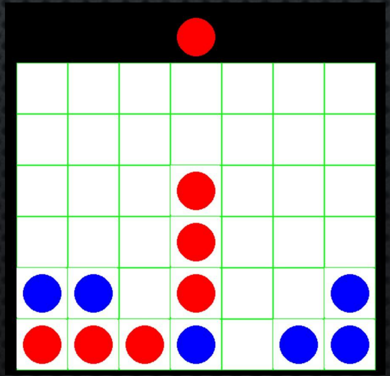
MODE 2 - Two Player Mode (Human Player vs Human Player)

MODE 3 - Training Mode (2 Different Reinforcement Learning Guided Computer Agents battle against each other over N iterations and learn to play the game)

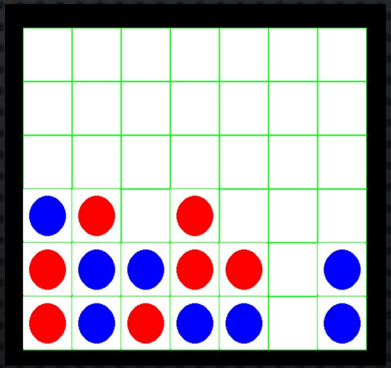




MODE1: Single Player Mode



MODE 2: Two PLAYER Mode



MODE 3: Training Mode

METHODOLOGY - ALGORITHMS

FOUR DIFFERENT ALGORITHMS WERE IMPLEMENTED TO MODEL THE COMPUTER AGENTS

MINIMAX AGENT
USES A BACKTRACKING, RECURSIVE ALGORITHM USED IN GAME THEORY TO MAKE MOVES THAT RESULT IN MAXIMUM IMMEDIATE GAIN

MONTE CARLO AGENT
USES REINFORCEMENT LEARNING TO LEARN DIRECTLY FROM GAME EXPERIENCES WITHOUT USING ANY PRIOR MARKOV DECISION PROCESS KNOWLEDGE

Q LEARNING AGENT
USES A REINFORCEMENT LEARNING OFF-POLICY VALUE BASED SCHEME BASED ON THE BELLMAN'S EQUATION TO LEARN THE VALUE OF OPTIMAL POLICY REGARDLESS OF ACTION

SARSA LEARNING AGENT
USES A REINFORCEMENT LEARNING ON-POLICY VALUE BASED SCHEME TO LEARN THE VALUE OF THE OPTIMAL POLICY BASED ON ACTION DERIVED FROM CURRENT POLICY

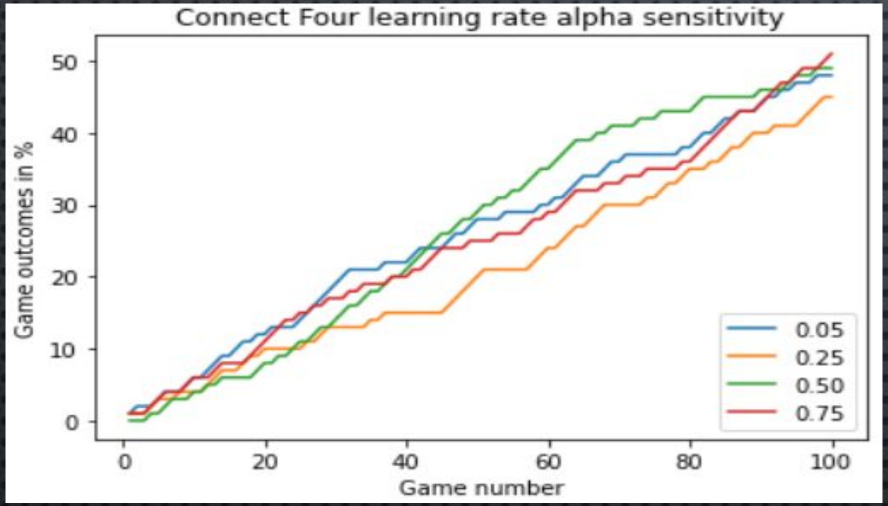
DURING TRAINING, THE FOUR AGENTS BATTLED AGAINST A COMPUTER AGENT THAT MADE RANDOM MOVES AND AGAINST EACH OTHER FOR N ITERATIONS, RESULTS ARE EVALUATED FOR WIN-RATE AND EFFICIENCY (AVG PLAY TIME)

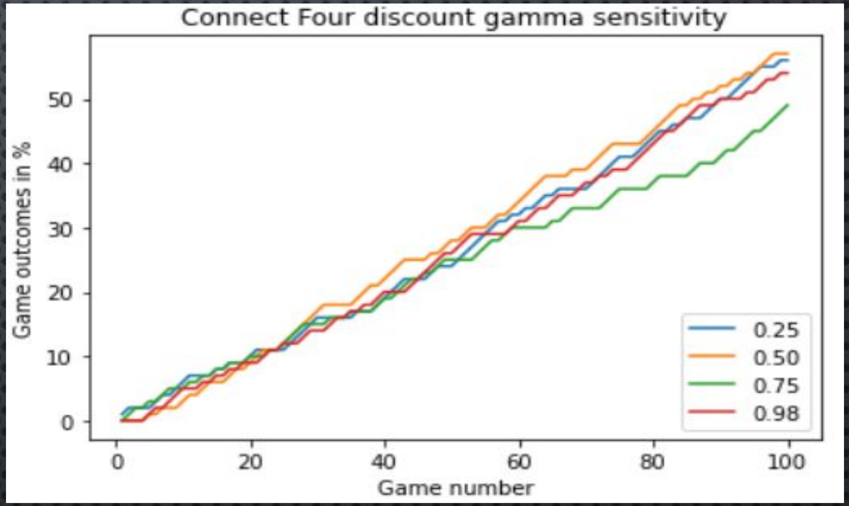
METHODOLOGY –
HYPER PARAMETER TUNING

ALGORITHM	HYPER PARAMETERS TUNED
Q LEARNING	alpha - [learning rate, tuning exp 0.05, 0.25, 0.3, 0.75, 0.9] gamma - [discount factor, tuning exp 0.25, 0.50, 0.75, 0.9, 0.98]
SARSA LEARNING	alpha - [learning rate, tuning exp 0.05, 0.25, 0.3, 0.75] gamma - [discount factor, tuning exp 0.25, 0.50, 0.75, 0.9, 0.98]
MONTE CARLO	exploration coefficient - [tuning exp 0.8, 1, 1.4, 1.6]
MINIMAX	Depth of recursion - 0.5

QLEARNER, SARSA LEARNER AGENTS – ARE FIRST TUNED FOR "LEARNING RATE" THEN TUNED FOR "DISCOUNT FACTOR" ON TOP

RESULTS – SENSITIVITY ANALYSIS OF QLEARNER AGENT





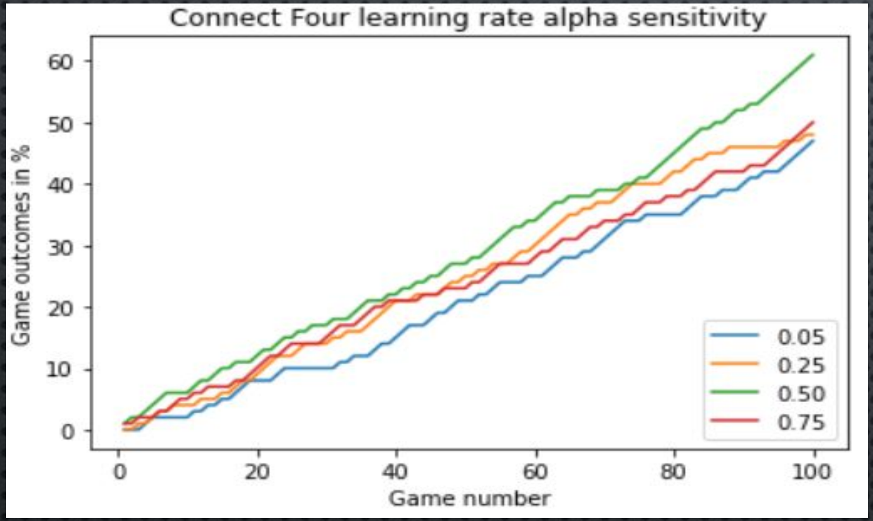
ALPHA/ LEARNING RATE (lr) SENSITIVITY ANALYSIS

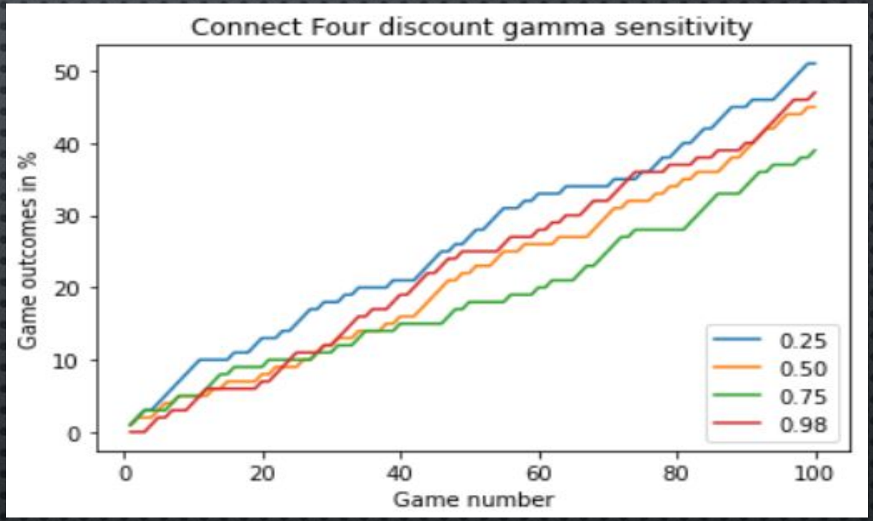
- For very lr = 0.05 learning happens rapidly initially then becomes more gradual
- For moderate lr = 0.50 learning happens gradually initially, then picks up to beat others **[best]**
- For high lr = 0.75 learning happens rapidly initially then slower than very lr

GAMMA/ DISCOUNT FACTOR SENSITIVITY ANALYSIS

- For low, moderate gamma = [0.25, 0.50] learning happens rapidly, highest win-rate **[best]**
- For high gamma = 0.75 learning happens gradually
- For very high gamma = 0.98 learning happens gradually initially then slows down

RESULTS – SENSITIVITY ANALYSIS OF SARSA LEARNER AGENT





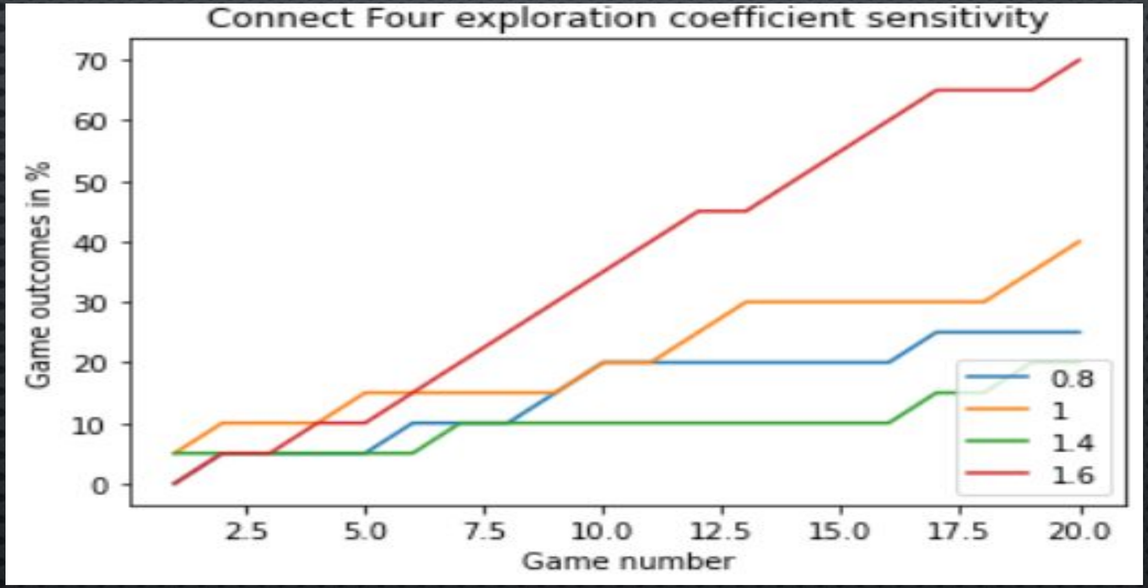
ALPHA/ LEARNING RATE (lr) SENSITIVITY ANALYSIS

- For very lr = 0.05 learning happens slowest
- For low, moderate lr = [0.25, 0.50] learning happens gradually initially, then picks **[best]**
- For high lr = 0.75 learning happens gradually, slower than low, moderate lr

GAMMA/ DISCOUNT FACTOR SENSITIVITY ANALYSIS

- For low gamma = 0.25 learning happens rapidly, highest win-rate **[best]**
- For moderate, very high gamma = [0.50, 0.98] learning happens gradually, slower than low gamma
- For high gamma = 0.75 learning happens the slowest

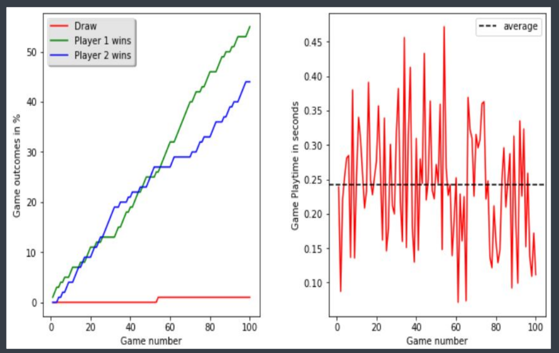
RESULTS – SENSITIVITY ANALYSIS OF MONTE CARLO AGENT



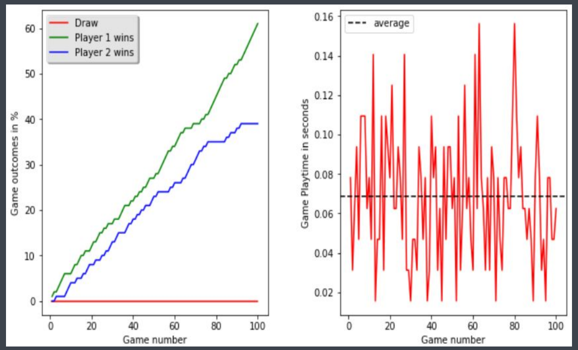
EXPLORATION COEFFICIENT [HOW MUCH MONTE CARLO TREE TO SEARCH] SENSTIVITY ANALYSIS

- Smaller Exploration Coefficient values lead to greater exploitation i.e., visited nodes are revisited
- Large Exploration Coefficient values lead to greater exploration i.e., new nodes are visited
- For exploration coefficient = 1.6, learning happens rapidly with high win-rate **[best]**

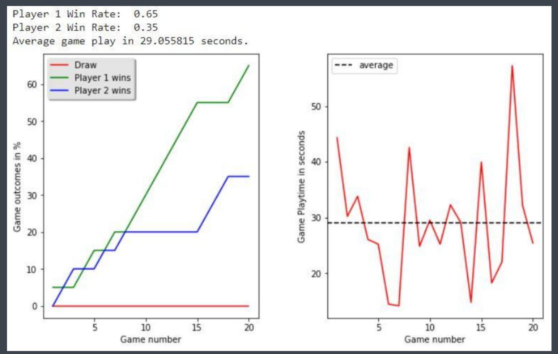
RESULTS – COMPUTER AGENTS VS RANDOM MOVE COMPUTER AGENT WIN RATE



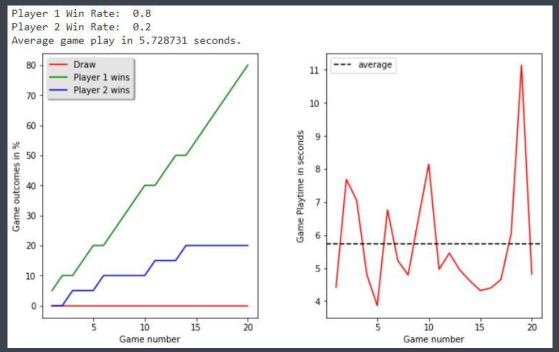
QLEARNER VS RANDOM MOVE AGENT
Win Rate 55%



SARSA LEARNER VS RANDOM MOVE AGENT
Win Rate 61%

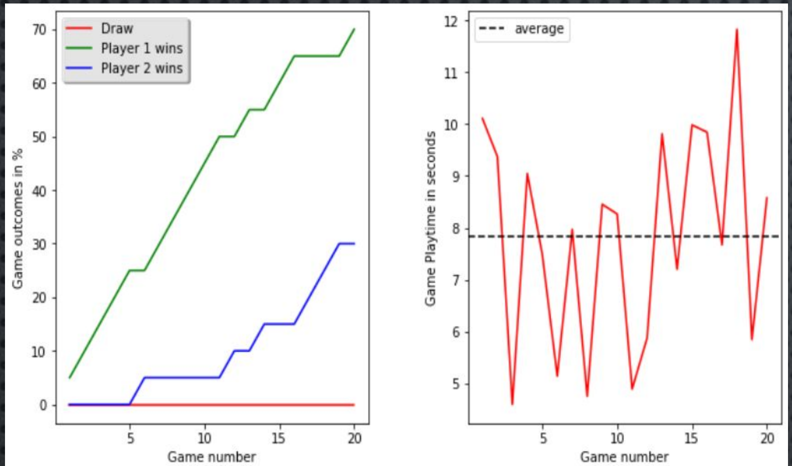


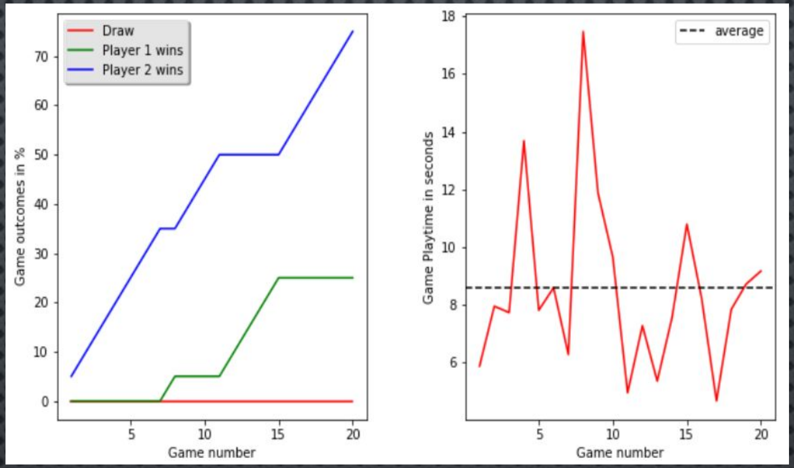
MONTE CARLO VS RANDOM MOVE AGENT
Win Rate 65%

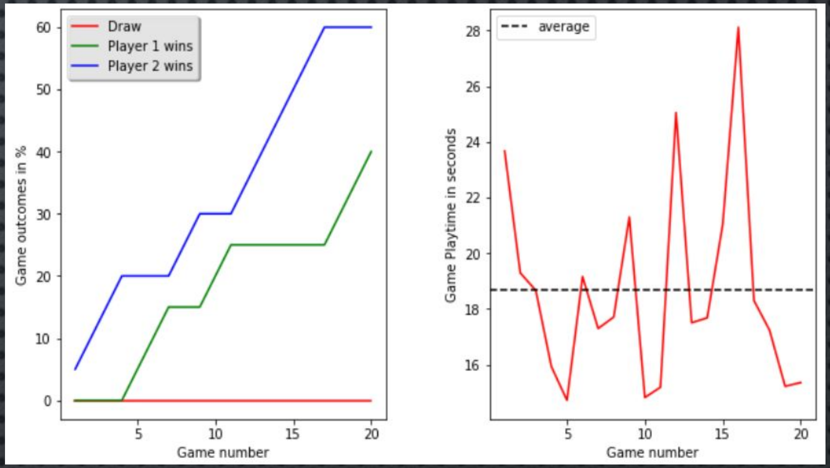


MINIMAX VS RANDOM MOVE AGENT
Win Rate 80%

MINIMAX has the highest Win Ratio = 80% in 20 battles vs RANDOM MOVE Agent







MINIMAX (Player 1) vs
Q LEARNER (Player 2)
QLEARNER Win Rate 30%

MINIMAX (Player 1) vs
SARSA LEARNER (Player 2)
SARSA LEARNER Win Rate 75%

MINIMAX (Player 1) vs
MONTE CARLO (Player 2)
MONTE CARLO Win Rate 60%

SARSA LEARNER has the highest Win Ratio = 75% in 20 battles vs MiniMax Agent

RESULTS – RL GUIDED COMPUTER AGENTS VS MINIMAX AGENT WIN RATE

WIN RATE COMPARISON

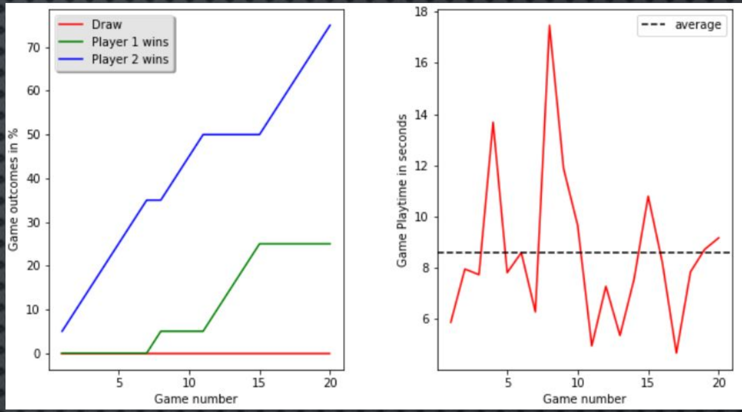
BEST OVERALL
WORST OVERALL
BEST IN COLUMN

	VS BASELINE AGENTS		VS RL GUIDED COMPUTER AGENT			**Overall Avg Win Rate**
	Random Move	Minimax *(80%-win rate vs Random Move)*	Q LEARNER	SARSA LEARNER	MONTE CARLO	
Q LEARNER	0.55	0.30	NA	0.44	0.35	0.41
SARSA LEARNER	0.61	0.75	0.55	NA	0.75	0.665
MONTE CARLO	0.65	0.60	0.65	0.25	NA	.5475

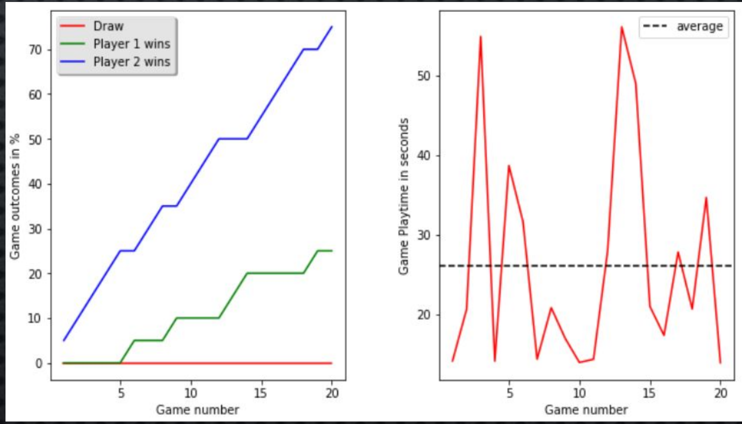
RL GUIDED COMPUTER AGENT VS BASELINE AGENTS
SARSA LEARNER HAS HIGHEST WIN RATE OF 75% VS MINIMAX AGENT

RL GUIDED COMPUTER AGENT VS RL GUIDED COMPUTER AGENTS
SARSA LEARNER HAS THE HIGHEST WIN RATE OF 75% VS MONTE CARLO

RESULTS – WIN RATE COMPARISON



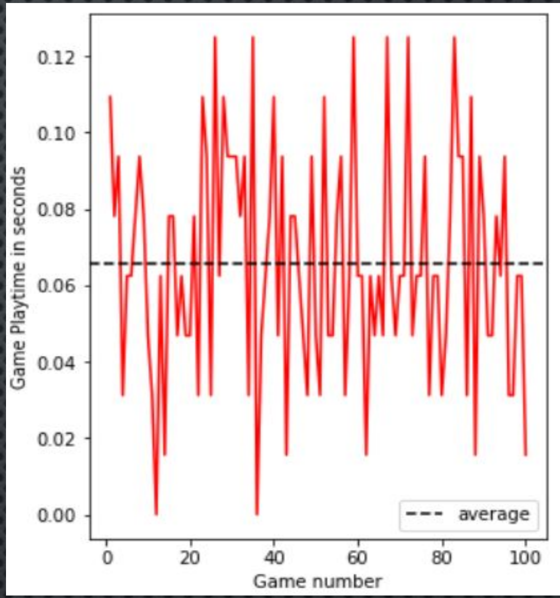
MINIMAX (Player 1) vs
SARSA LEARNER (Player 2)
SARSA LEARNER Win Rate 75%



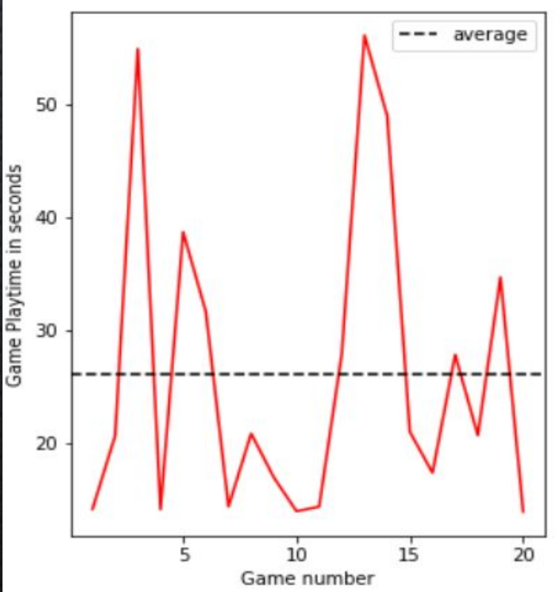
MONTE CARLO (Player 1) vs
SARSA LEARNER (Player 2)
SARSA LEARNER Win Rate 75%

SPEED COMPARISON

FASTEST SLOWEST	VS BASELINE AGENTS		VS RL GUIDED COMPUTER AGENT		
	Random Move	Minimax *(80%-win rate vs Random Move)*	Q LEARNER	SARSA LEARNER	MONTE CARLO
Q LEARNER	0.2422	**7.835**	NA	**0.0634**	24.138
SARSA LEARNER	**0.0681**	8.575	0.0634	NA	**26.117**
MONTE CARLO	**29.055**	**18.661**	**24.138**	**26.117**	NA



S A R S A



M O N T E C A R L O

RL GUIDED COMPUTER AGENT VS BASELINE AGENTS

SARSA LEARNER IS FASTEST, AVG PLAY TIME OF 0.06s VS RANDOM MOVE AGENT

MONTE CARLO IS SLOWEST, AVG PLAY TIME OF 29.06s VS RANDOM MOVE AGENT

RL GUIDED COMPUTER AGENT VS RL GUIDED COMPUTER AGENTS

SARSA LEARNER IS FASTEST, AVG PLAY TIME OF 0.0634s VS Q LEARNER AGENT

MONTE CARLO IS SLOWEST, AVG PLAY TIME OF 26.117s VS SARSA LEARNER AGENT

RESULTS – EFFICIENCY COMPARISON

GITHUB REPO