Two-level reinforcement learning model in competitive games

Team 14: Daniel McGartland, Jakob Smith, Anthony Solorio, Sponsor: Mark V. Albert and Chengping Yuan
Department of Computer Science & Engineering, University of North Texas and Biomedical AI Lab, University of North Texas (biomed-AI.com)

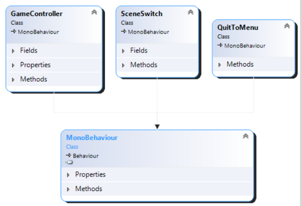
Overview

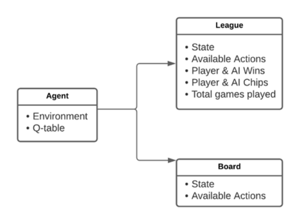
Reinforcement learning has been used extensively to learn the low-level tactical choices during gameplay; however, less effort has been invested in the strategic decisions of when and how to effectively engage a diverse set of opponents. Here we implement a two-tier reinforcement learning model to play competitive games and effectively engage in matches with different opponents to maximize earnings. The multi-agent environment has four types of learners, which vary in their ability to make board-level decisions (tactics) and their ability to bet or withdraw from game play (strategy). The agents are implemented in two different competitive games: Connect 4 and Tic-Tac-Toe. A human can play either a single match against a selected difficulty agent that makes no strategic choices or a series of league matches against a randomly chosen agent that will vary in its tactical and strategic ability.

Features

- Play Tic-Tac-Toe and Connect 4 against AI on a mobile device.
- Win, lose, and tie games
- Choose difficulty level of AI for single match games
- Reset the game to play against the same AI again
- Play against multiple AI of differing levels in league games
- Make bets and win chips based off of performance in league games
- Return to menu and switch games on the fly
- Compatible with both Android and iOS

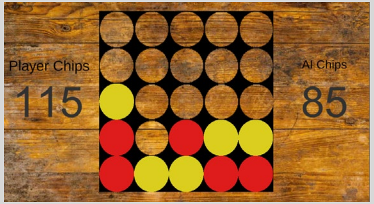
Design

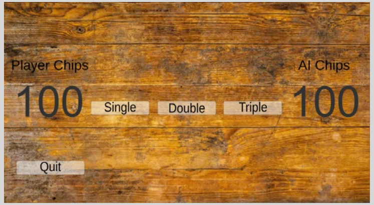




Technologies

Screenshots





Testing

Because our two-level reinforcement learning model is expressed in the form of a pair of competitive games our primary focus for testing was playability.

To that end we extensively tested the various difficulty levels for each AI. By having the reinforcement learning AI repeated play one of the games against itself it could learn what moves do and do not work in certain situations storing that information into a Q-table (shown below) which could then be imported into the game's AI. In this way an AI that had played more games would be more difficult to defeat.

To test these difficulties, we had various members of our team play against these Q-table backed AI players to determine how many games an AI should have played for each difficulty. For example: in Tic-Tac-Toe we came to the conclusion that an AI that had played 40,000 games was sufficient for an easy difficulty whereas the hard difficulty would need to play 100,000 times in order to be nearly unbeatable.

$$\boxed{Q^{\pi}(s_t, a_t)} = \boxed{\underline{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots}\boxed{s_t, a_t}]$$

Q-Values for the state given a particular state

Expected discounted cumulative reward

Given the state and action

Q represents the quality of a particular move

Q-Learning allows the AI to choose the action that will maximize its chances to win from experience

Additional Resources

- biomed-AI.com/news