**Description**

Our project implementation is completely working as mentioned in the project description.

The Phase II internal design is same as the phase I, we changed some heuristics according to constraints given in Phase II. We start our game by choosing 14 cards randomly, 3 adversaries (default value but configurable), valid three cards, and gods rule expression. Each player gets a chance to play their turn one after other in clockwise direction. On the adversaries' turn, they randomly play a card and on the agent's turn, it selects a card with positive or negative result to maintain the uniform distribution of result values (true/false), from our rule expression. We created the rule from decision tree built using board state and gain Ratio as heuristic to choose best attribute to split.

Any player can stop the game. Adversary stops the games after certain number of plays between 20 and 200, chosen randomly at start of the game. Our agent returns the best rule that It has when adversary stops the game. If no other adversary has returned their rule yet, our agent decides to end the game by returning the rule based on confidence measure which we calculate using number of consecutive correct guesses we made i.e. 50.

To check whether our rule expression is equivalent to god rule, we are checking all possible 52\*52\*52 triplets with our rule and god’s rule, if both are same for all combinations then we are considering our rule is equivalent to god’s rule, and we print the percentage of how close our rule is to gods rule.

**How to Run**

*import New\_Eleusis\_Agent\_Phase2 as ag ag.play\_game(card1, card2, card3, gods\_rule\_expression)*

card1, card2, card3 are initial valid three cards and gods\_rule\_expression is gods rule

**Understanding output** *adversary limits: [107, 58, 144]* adversaries want to stop the game at 107, 58, 144 plays respectively *card[0]: we play the 7S* our agent return 7S card at 0th play. *card[1]: adv[0] plays 7C* adversary 0 returned the 7C card at 1th play. *Adversary 0 has audaciously decided to return a rule!* Adversary 0 stopped the game *Our rule: equal(is\_royal(current),True)* Our final rule at end of the game *Testing exact logical equivalency with all 3-card tuples...* Now our agent testing logical equivalency of our rule and gods rule, please be patient for a while *Our rule guesses 100.0% correctly* Our rule is 100% similar to gods rule *Our score before end of game bonuses: 2* score of our agent before calculating the logical equivalency bonus *Final, awesome, score: -73* Final score for the game is