

AI Poker Tutorial

Table of contents

Home	4
I Games	5
1 Expected Value Example: Poker	6
1.1 Section 1	6
1.2 Section 2	6
1.2.1 Section 2.1	6
1.2.2 Section 2.2	6
1.3 Section 3	6
1.3.1 Section 3.1	6
1.3.2 Section 3.2	6
1.3.3 Section 3.3	6
1.4 Section 4	6
2 Kuhn Poker	7
3 Blackjack	8
4 Tic Tac Toe	9
5 Leduc Poker	10
6 Rock Paper Scissors	11
7 Texas Hold'em	12
8 Texas Tac Toe	13
9 Rock Poker Scissors	14
II Game Theory	15
10 Best Response	16

11 Game Theory Foundations	17
12 Best Response	18
13 Game Trees	19
 III Reinforcement Learning	 20
14 Counterfactual Regret Minimization (CFR)	21
15 Monte Carlo Methods	22
16 Reinforcement Learning	23
 IV Optimal Strategies	 24
17 Kuhn Poker CFR	25
18 Kuhn Poker	26
19 MCCFR (Monte Carlo CFR)	27
 V Exploitative Strategies	 28
20 Best Response	29
21 Opponent Modeling	30
 VI Abstracting Large Games	 31
22 Card Abstractions	32
23 Bet Abstractions	33
 VII Setup	 34
24 Intro and Poker Foundations	35
25 Poker Camp Servers	36
26 Agents	37

Home

AI Poker Camp Beta Course Materials

Visit <https://poker.camp> for info on the course

Part I

Games

1 Expected Value Example: Poker

1.1 Section 1

1.2 Section 2

1.2.1 Section 2.1

1.2.2 Section 2.2

1.3 Section 3

1.3.1 Section 3.1

1.3.2 Section 3.2

1.3.3 Section 3.3

1.4 Section 4

2 Kuhn Poker

This is the CFR page

$$E = mc^2$$

3 Blackjack

This is the CFR page

$$E = mc^2$$

4 Tic Tac Toe

This is the CFR page

$$E = mc^2$$

5 Leduc Poker

This is the CFR page

$$E = mc^2$$

6 Rock Paper Scissors

This is the CFR page

$$E = mc^2$$

7 Texas Hold'em

This is the CFR page

$$E = mc^2$$

8 Texas Tac Toe

This is the CFR page

$$E = mc^2$$

9 Rock Poker Scissors

This is the CFR page

$$E = mc^2$$

Part II

Game Theory

10 Best Response

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

11 Game Theory Foundations

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

12 Best Response

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

13 Game Trees

This is the CFR page

$$E = mc^2$$

Part III

Reinforcement Learning

14 Counterfactual Regret Minimization (CFR)

This is the CFR page

$$E = mc^2$$

15 Monte Carlo Methods

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

16 Reinforcement Learning

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

Part IV

Optimal Strategies

17 Kuhn Poker CFR

This is the CFR page

$$E = mc^2$$

18 Kuhn Poker

This is the Kuhn Poker page

19 MCCFR (Monte Carlo CFR)

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

Part V

Exploitative Strategies

20 Best Response

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

21 Opponent Modeling

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

Part VI

Abstracting Large Games

22 Card Abstractions

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

23 Bet Abstractions

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

Part VII

Setup

24 Intro and Poker Foundations

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

25 Poker Camp Servers

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

26 Agents

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.