AI Final Project:
Texas Hold'em Casino

Due: 2022/06/17 11:59 PM

No late submission is allowed

Contact: fai@csie.ntu.edu.tw

Outline

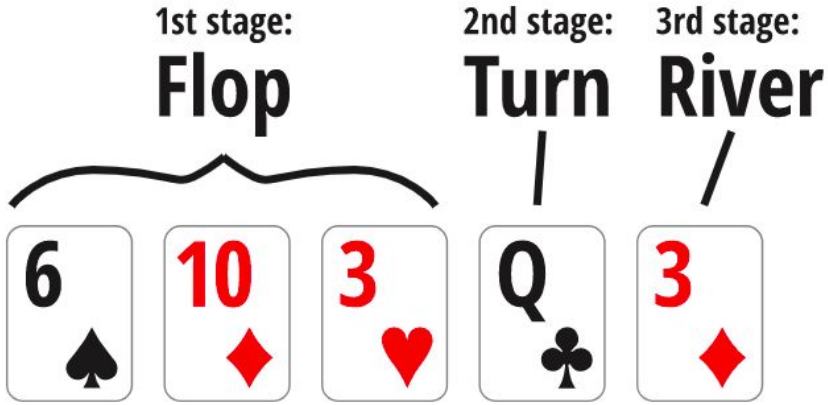
- Introduction
- Texas Hold’em Rules
- Game Parameters in Final Project
- Environment
- Sample Code Explanation
- Competition Rules
- Submission
- Grading
- Questions

Introduction

In this project, we will implement Texas Hold’em AI to compete in the CSIE casino, try your best to create the most intelligent agent to win the game!

Texas Hold'em Rules –Heads Up

- In Texas hold'em, each player is dealt two cards face down (the 'hole cards')
- The face-up cards are called the 'community cards.'
- Each player is free to use the community cards in combination with their hole cards to build a five-card poker hand. (Build best 5 combination out of 7 cards)
-



5 community cards

Texas Hold’em Rules –Heads Up

- A game of Texas hold'em features several betting rounds
- Players get two private and up to five community cards

The blinds are forced to bets

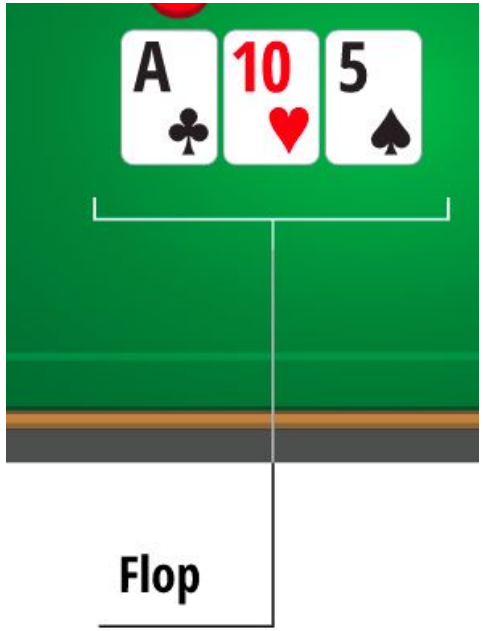
- Before receiving any card, the first player should post a “small blind”
- The 2nd player should post a “big blind”.
- Big blind is two times larger than small blind.

Texas Hold'em Rules –Heads Up

- **The first Betting Round: Preflop**
- The player has 3 options
  - **Call**: match the amount of the big blind
  - **Raise**: increase the bet within the specific limits of the game
  - **Fold**: throw the hand away. If the player chooses to fold, he or she is out of the game and no longer eligible to win the current hand.

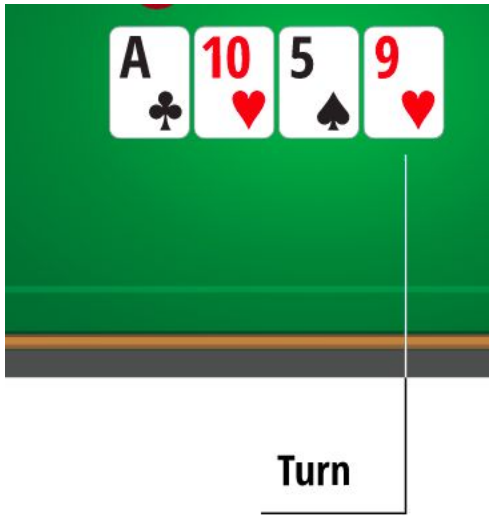
Texas Hold'em Rules –Heads Up

- **The Second Betting Round: The Flop**
- After the first preflop betting round has been completed, the first three community cards are dealt
- A second betting round follows involving only the players who have not folded already.
- Players can only choose action “call”, “fold”, or “raise”



Texas Hold'em Rules –Heads Up

- **Third Betting Round: The Turn**
- The fourth community card, called the turn, is dealt face-up following all betting action on the flop.
- Once this has been completed, another round of betting occurs, similar to that on the previous street of play.
- Players can only choose action “call”, “fold”, “raise”



Texas Hold'em Rules –Heads Up

- **Final Betting Round: The River**
- The fifth community card, called the river, is dealt face-up following all betting action on the turn.
- Once this has been completed, another round of betting occurs, similar to what took play on the previous street of play.

Texas Hold'em Rules –Heads Up

- **The Showdown**
- After all betting action has been completed, the remaining players in the hand with hole cards now expose their holdings to determine a winner.
-

The player with the best combination of five cards wins

Texas Hold'em Rules –Heads Up

- **The Hands in Texas Hold'em**

- **Royal Flush** — five cards of the same suit, ranked ace through ten; e.g., A♥ K♥ Q♥ J♥ 10♥
- **Straight Flush** — five cards of the same suit and consecutively ranked; e.g., 9♣ 8♣ 7♣ 6♣ 5♣
- **Four of a Kind** — four cards of the same rank; e.g., Q♣ Q♥ Q♦ Q♠ 4♦
- **Full House** — three cards of the same rank and two more cards of the same rank; e.g., J♣ J♥ J♠ 8♦ 8♥

Texas Hold'em Rules –Heads Up

- **Flush** — any five cards of the same suit; e.g., A♠ J♠ 8♠ 5♠ 2♠
- **Straight** — any five cards consecutively ranked; e.g., Q♣ J♦ 10♥ 9♠ 8♦
- **Three of a Kind** — three cards of the same rank; e.g., 8♣ 8♠ 8♦ K♣ 4♥
- **Two Pair** — two cards of the same rank and two more cards of the same rank; e.g., A♠ A♣ J♦ J♣ 7♠
- **One Pair** — two cards of the same rank; e.g., 10♥ 10♣ 9♥ 4♦ 2♦
- **High Card** — five unmatched cards; e.g., A♣ J♦ 10♠ 5♣ 2♥ would be called "ace-high"

Game Parameters in Final Project

- Heads up competition, the player who have more money left is the winner.
- Max round of game: 20
- Initial stack for each player: 1000
- Small blind: 5
- There is no upper bound for “raise”, you can all in all your money at once.
- You have to raise action within **5** seconds in every turn, or you will be seen as declaring “fold”.

Environment

- Setup python version, on *linux{1~15}.csie.org* workstation

  pyenv install -v 3.8.13

- Switch to the installed python 3.8.13

  pyenv global 3.8.13

- Install the packages

  pyenv exec pip install -r requirement.txt

- Run the ***start_game.py*** to see how the game works

  pyenv exec python start_game.py

Environment

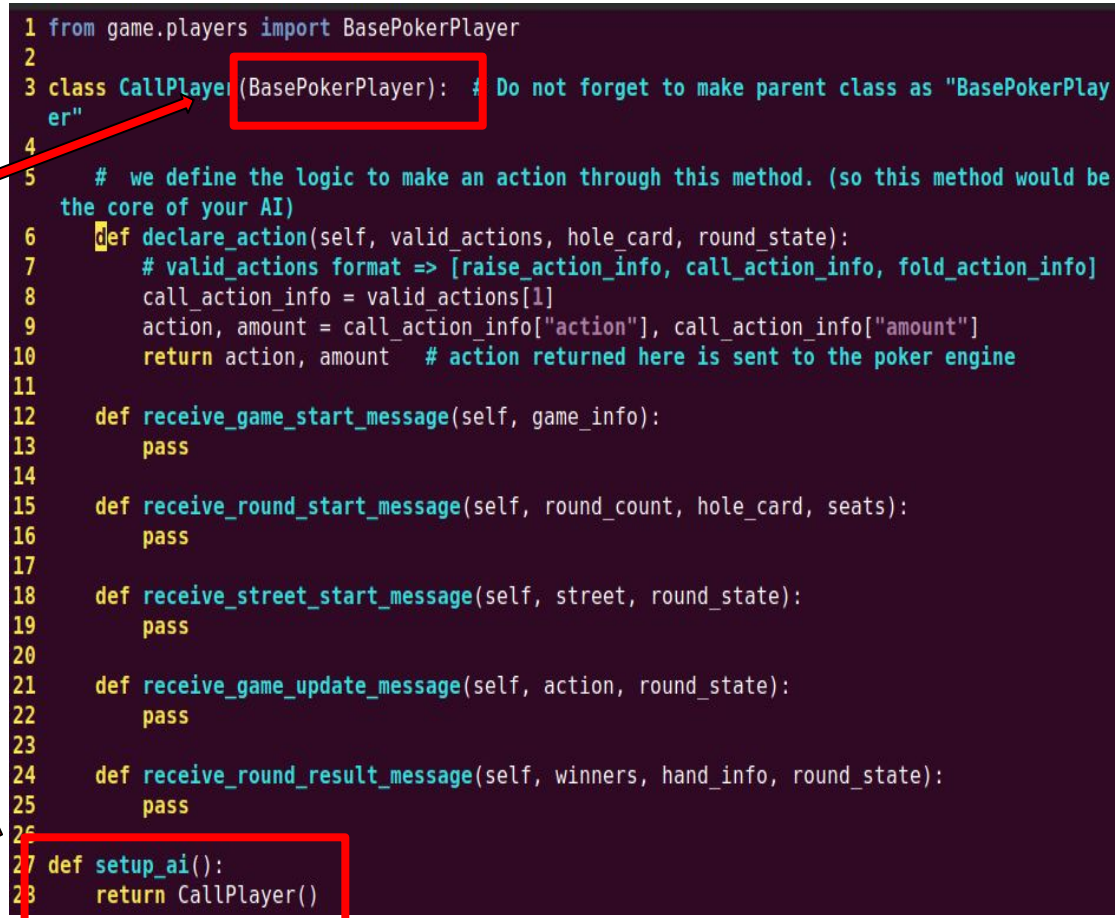
- python 3.8.13
- numpy==1.22.3
- torch==1.11.0
- scikit-learn==1.0.2
- tensorflow==2.8.0
- keras==2.8.0
- pytorch_lightning==1.6.1
- tqdm==4.64.0
- If you want to use other packages, please email the TA first.

Sample Code Explanation

- After you unzip the sample code, you will see the directory with the following structure
- final_project/
  - |--- start_game.py *<the code to help you test locally>*
  - |--- requirement.txt *<python package needed in this project>*
  - |--- baseline0.cpython-38-x86_64-linux-gnu.so *<the binary file that can be import in csie workstation>*
  - |--- game/ *<it contains all needed game objects, you should not modify any file in this directory>*
  - |--- agents/ *<it contains sample agents for you to play with>*
    - |--- call_player.py *<the agent always to “call” action>*
    - |--- random_player.py *<the agent choose action randomly>*
    - |--- console_player.py *<you can play the game in interaction mode with this player>*

Sample Code Explanation

- Your agent class should make parent class as “BasePokerPlayer”
- You should override 7 functions
  - declare_action
  - receive_game_start_message
  - receive_round_start_message
  - receive_street_start_message
  - receive_game_update_message
  - Receive_round_result_message
- You should include a function named “setup_ai” that return your agent class



```
from game.players import BasePokerPlayer

class CallPlayer(BasePokerPlayer):  # Do not forget to make parent class as "BasePokerPlayer"

    #  we define the logic to make an action through this method. (so this method would be the core of your AI)
    def declare_action(self, valid_actions, hole_card, round_state):
        # valid_actions format => [raise_action_info, call_action_info, fold_action_info]
        call_action_info = valid_actions[1]
        action, amount = call_action_info["action"], call_action_info["amount"]
        return action, amount   # action returned here is sent to the poker engine

    def receive_game_start_message(self, game_info):
        pass

    def receive_round_start_message(self, round_count, hole_card, seats):
        pass

    def receive_street_start_message(self, street, round_state):
        pass

    def receive_game_update_message(self, action, round_state):
        pass

    def receive_round_result_message(self, winners, hand_info, round_state):
        pass

def setup_ai():
    return CallPlayer()
```

Sample Code Explanation

- To test your agent locally, you can use "*python3 start_game.py*"

1. Import setup_ai function for every agent
2. Setup game configuration with predefined rules
3. Register users with the agent
4. Play the game and get the result

```
1 from game.game import setup_config, start_poker
2 from agents.call_player import setup_ai as call_ai
3 from agents.random_player import setup_ai as random_ai
4 from agents.console_player import setup_ai as console_ai
5
6 config = setup_config(max_round=20, initial_stack=1000, small_blind_amount=5)
7 config.register_player(name="p1", algorithm=call_ai())
8 config.register_player(name="p2", algorithm=random_ai())
9 config.register_player(name="me", algorithm=console_ai())
10 game_result = start_poker(config, verbose=1)
```

Sample Code Explanation

- To play the game interactively, you can try agent in ”console_player.py”, it provide interactive agent that allow you to play the game step by step.
- However, It is not allow to use the interactive agent to play with the baseline, you can only test it locally.

```
1  from game.game import setup_config, start_poker
2  from agents.call_player import setup_ai as call_ai
3  from agents.random_player import setup_ai as random_ai
4  from agents.console_player import setup_ai as console_ai
5
6  config = setup_config(max_round=20, initial_stack=1000, small_blind_amount=5)
7  config.register_player(name="p1", algorithm=call_ai())
8  config.register_player(name="p2", algorithm=random_ai())
9  config.register_player(name="me", algorithm=console_ai())
10 game_result = start_poker(config, verbose=1)
```

Sample Code Explanation

- Example result after submit to baseline server.
- Game Rules
- Game result after playing 20 runs
- p1 won in this example

Sample Code Explanation

- We will provide 3 baseline in different difficulty in this project ,and we will not release not implementation details about those baseline agents.
- The baseline will be in binary executable format compatible in csie workstation.
  - <your student id>@linux{1~15}.csie.org
- If you don't have csie workstation account, please refer to the link
- We will release baseline agents **3** weeks before the deadline. For now, we only provide "baseline0". It is for test only and does not account for any credit.

```
6  from game.game import setup_config, start_poker
7  from agents.random_player import setup_ai as random_ai
8  from baseline0 import setup_ai as baseline0_ai
9
10 config = setup_config(max_round=20, initial_stack=1000, small_blind_amount=5)
11 config.register_player(name="p1", algorithm=baseline0_ai())
12 config.register_player(name="p2", algorithm=random_ai())
13 game_result = start_poker(config, verbose=1)
```

Competition Rules

- All matches, including the matches against baseline agents, are “best of five” (each game consists of 20 rounds)
- We will held heads up competition for the whole class after the deadline.
- Initially, students will be randomly assigned into a group of 6 people.
- Everyone will have 5 opponents in total.
- You will get 2 points if you win one competition, and you can get maximum 10 points in the competition part if you win all the matches.
- **Bonus**
  - We will select the top 32 students who have the most money left to go through the single elimination tournament.
  - The top 4 students will be granted bonus points for the final grades
    - 1st: 4 pt
    - 2nd: 3 pt
    - 3rd: 2 pt
    - 4rd: 1 pt

Report

- Your report should include but not limited to
  - The methods you have tried.
  - Your configurations(e.g. hyperparameters)
  - Comparison of your methods.
  - Provide some discussion and conclusion.
  - What method you choose to submit finally.
- You should write your report in maximum 4 A4 pages in pdf format.
- The grading of the report will base on the number of method you tried, completeness, novelty, and clarity of writing.

Submission

- We will evaluate both the functionality of the **code** and the quality of the **report**.
- You have to compress your agent and the related files in a single .zip named with your student id in lowercase.
- Example:
  - b0x902xxx.zip
  - b0x902xxx/
    - |-- report.pdf
    - |-- src/
      - |-- agent.py
      - |-- other related file needed

Grading

- Report (60 pt)
- Beat the baselines (30 pt)
  - baseline1 (10 pt)
  - baseline2 (10 pt)
  - baseline3 (10 pt)
- Competition (10 pt)
  - You will get 2 points if you win one competition, and you can get maximum 10 points in the competition part if you win all 5 matches.
- Bonus
  - We will select the top 32 students who have the most summation of money left in the games to go through the single elimination tournament.
  - The top 4 students will be granted bonus points for the final grades
    - 1st: 4 pt
    - 2nd: 3 pt
    - 3rd: 2 pt
    - 4rd: 1 pt
  -

Questions?

- Please contact fai@csie.ntu.edu.tw