**Group Members**

Rich Jiang rijiang

Pavan Kumar Vaddineni, pvaddine

Chaitanya Mandava, cmandava

Wanning Ding, wding04

**Link to the github repository**

<https://github.com/rjiang1/StockTrade.git>

**Game Rules**

**States**

The state of the game at each turn is the number of the shares of stock the player holds, the cash player has currently and the current day's information of the stock AAPL.

**State Representation**

The player states are represented as class Player which has attributes portfolio and cash.

The stock states are represented as class Stock which has attributes ticker and price.

**Valid Actions**

* Buy - Enter B/b to buy a stock and then the player has to enter number of stocks he/she want to buy, one cannot buy stock worth more than his/her cash balance.
* Sell - Enter S/s to sell a stock and then the player have to enter number of stocks he/she want to sell, one cannot sell stocks whose count is more than what he/she actually owns.
* Do Nothing/ Hold - Enter H/h, the player will be shown next day apple stock.
* Exit - Enter E/e the player will be exited out of prediction game and his/her portfolio, profit/ loss will be shown.

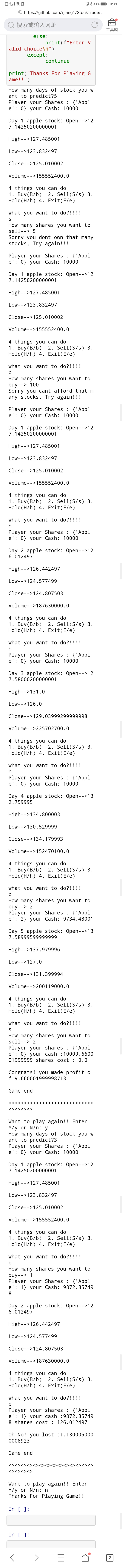
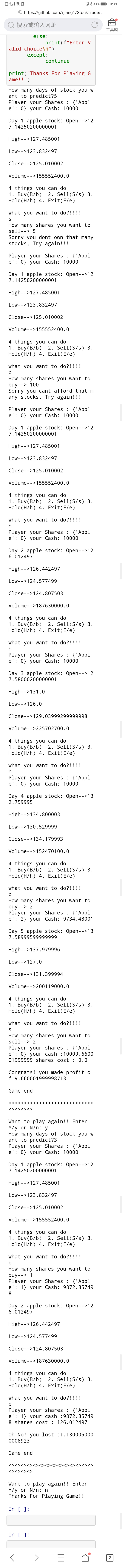
**Game Over**

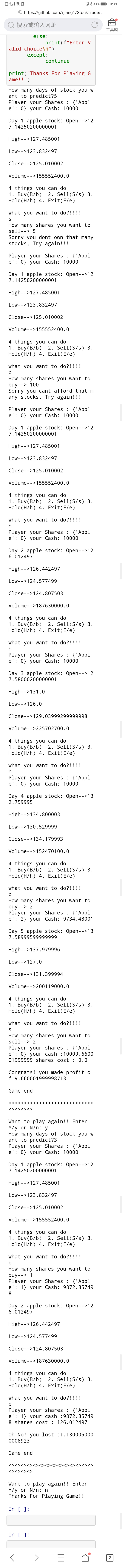
The game is over after the n days player’s prediction(n is what the player gave as input at first) or when the player enters the action Exit.

**Winner/score Determined**

TotalCash will be calculated as cash Balance + total shares cost that one holds according to n+1 day apple stock open price. And the player ends up in profit [win] i.e., totalCash > 10000 (starting balance) or loss i.e., totalCash < 10000.

**Example**



**Game 1:**

Want to predict 5 days of the stock transaction.

**Day 1:**

Try sell action, Fail.

Try buy 100 shares action, Fail.

Try hold action,

Return shares : {'Apple': 0} cash: 10000.

**Day 2:**

Try hold action,

Return shares : {'Apple': 0} cash: 10000.

**Day 3:**

Try hold action,

Return shares : {'Apple': 0} cash: 10000.

**Day 4:**

Try buy 2 shares action,

Return shares : {'Apple': 2} cash: 9734.48001.

**Day 5:**

Try sell 2 shares action,

Return shares : {'Apple': 0} cash :10009.660001999999

shares cost : 0.0.

**The player win the game** and made profit of: 9.660001999998713.

Game is over.

**Game 2:**

Want to predict 3 days of the stock transaction.

**Day 1:**

Try buy 1 share action,

Return shares : {'Apple': 1} cash: 9872.857498.

**Day 2:**

Try exit action,

Return shares : {'Apple': 1} cash :9872.857498

shares cost : 126.012497.

**The player loss the game** and lost: 1.1300050000008923.

Game is over.

**Problem extension**

We varied the size of the problem instance in our experiments by letting the player set the number of days he/she wants to predict the stock transaction.

For the actions, we have added an exit action in the first version of the experiment, which allows players to exit the game at any time.

And for the different numbers of players, The problem can be extended to two, three players or more players. They all predict according to the same stocks’ information and the player who ends up in the highest profit win the game.

For the different numbers of stocks, we can add more stocks’ information except AAPL, like GOOG, MSFT, etc in the future. Then when the players do the buy, sell or hold action, players need to not only set the number of shares but also choose which stock to trade.

**Language and Libraries**

Python, numpy, pandas, pytorch/tensorflow/keras