TÉCNICO LISBOA

Object Oriented Programming

Project Report

Grupo #39

- 67568 - Daniel Mendes
- 67724 - Tiago Barral
- 75352 - Filipe Novais

Lisbon
May 14, 2017

1 Statistics

In order to test our *hand evaluator* we ran the program with a simple *for loop* over the whole main one million times. This was made by serializing a class to a file so that it could retrieve the object/class in a next iteration of the program, updating the information. The results are shown in table 1. By comparing this with the theoretical percentages[1] we can have a better understanding of the *hand evaluator*'s accuracy for each hand. Looking at the results we could perhaps have better results but the overall seems to approve the hand evaluator.

	Theoretically		**In Pratice**			
Hand	**Odds**	**Odds(%)**	**Number of occurrences**	**Percentage**	**Deviation(%)**	
Royal Flush	1 in 649740	153.917e-6	1	100e-6	0.3503	
Straight Flush	1 in 72193	1.3852e-3	15	1.5e-3	8.2876	
Four of a Kind	1 in 4165	24.009e-3	231	23.1e-3	3.7861	
Full House	1in 694	144.09e-3	1439	143.9e-3	0.1318	
Flush	1 in 509	196.46e-3	1962	196.2e-3	0.1323	
Straight	1 in 255	392.16e-3	4049	404.9e-3	3.2487	
Three of a Kind	1 in 47	2.1276	21203	2.1203	0.343	
Two Pairs	1 in 21	4.7619	46931	4.6931	1.4448	
Jacks or Better	1 in 7.69	13.0039	133341	13.3341	2.539	
# of hands			1000000			

Table 1: General statistics for testing accuracy of hand evaluator.

One aspect about the Statistics is that, only the hand chosen by the player is accounted. For instance, if the player gets a royal flush but decides not to hold any card and gets a pair in the drawing hand, only the pair is accounted.

2 PayOutTable

The search for the amount to pay is made within a simple 11x6 matrix (int[][]) for its dimensions were not sufficient to justify the use of an *hashmap.*

3 Modes

In Interactive mode the program has one more feature to close the program. The command *q* (quit) terminates the program. However, it is only possible to use it between bets. This feature is also incorporated in Debug mode where the program is terminated after all the commands have been ran. In Simulation mode the program terminates after the number of deals has been completed.

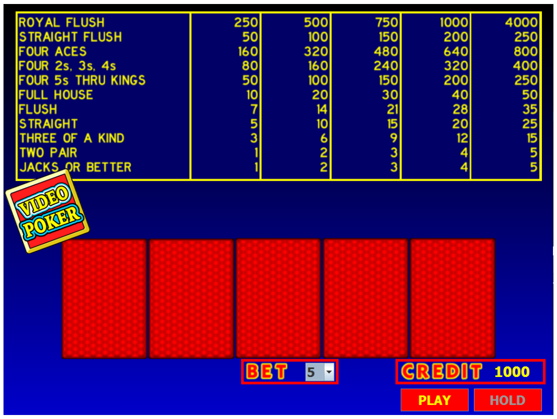
We added one extra mode, the Test mode so that we could test the advice class in a easier way by picking the cards for the hand, without need for credit.

[1] http://www.videopokerinfo.com/handodds.htm

This way we were able to understand if the advice was working correctly and made the job much faster. The Debug mode could have been used but writing the hand in the file would have made the chore much slower.

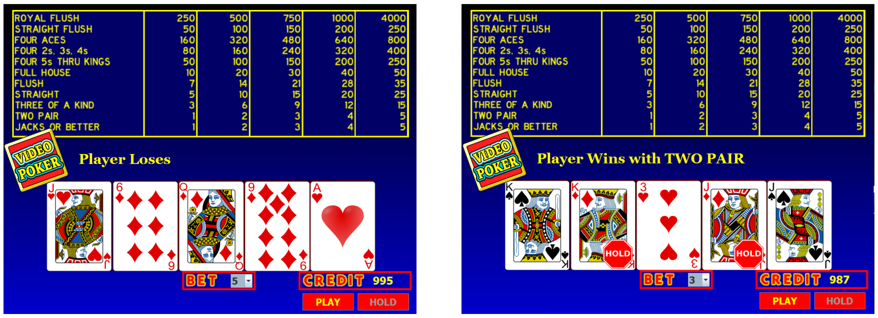
4 User Interface

The user interface was made using SWING GUI. The initial balance of the user is default to be 1000 credits but it can be easily changed. In figures 1 and 2 there are several printscreens of the GUI.



ROYAL FLUSH	250	500	750	1000	4000
STRAIGHT FLUSH	50	100	150	200	250
FOUR ACES	160	320	480	640	800
FOUR 2s, 3s, 4s	80	160	240	320	400
FOUR 5s THRU KINGS	50	100	150	200	250
FULL HOUSE	10	20	30	40	50
FLUSH	7	14	21	28	35
STRAIGHT	5	10	15	20	25
THREE OF A KIND	3	6	9	12	15
TWO PAIR	1	2	3	4	5
JACKS OR BETTER	1	2	3	4	5

Figure 1: Graphic User Interface.



ROYAL FLUSH	250	500	750	1000	4000
STRAIGHT FLUSH	50	100	150	200	250
FOUR ACES	160	320	480	640	800
FOUR 2s, 3s, 4s	80	160	240	320	400
FOUR 5s THRU KINGS	50	100	150	200	250
FULL HOUSE	10	20	30	40	50
FLUSH	7	14	21	28	35
STRAIGHT	5	10	15	20	25
THREE OF A KIND	3	6	9	12	15
TWO PAIR	1	2	3	4	5
JACKS OR BETTER	1	2	3	4	5

Player Loses

(a) GUI after the deal.

ROYAL FLUSH	250	500	750	1000	4000
STRAIGHT FLUSH	50	100	150	200	250
FOUR ACES	160	320	480	640	800
FOUR 2s, 3s, 4s	80	160	240	320	400
FOUR 5s THRU KINGS	50	100	150	200	250
FULL HOUSE	10	20	30	40	50
FLUSH	7	14	21	28	35
STRAIGHT	5	10	15	20	25
THREE OF A KIND	3	6	9	12	15
TWO PAIR	1	2	3	4	5
JACKS OR BETTER	1	2	3	4	5

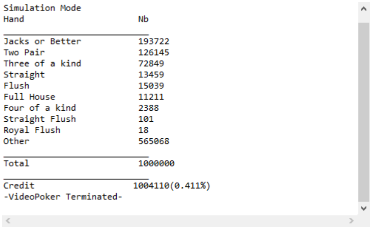
Player Wins with TWO PAIR

(b) GUI after the hold.

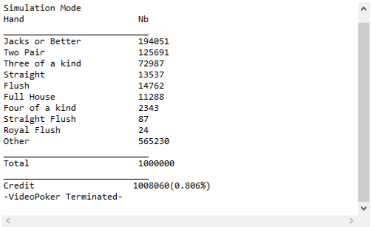
Figure 2: Different stages of GUI.

5 Advice and Strategy

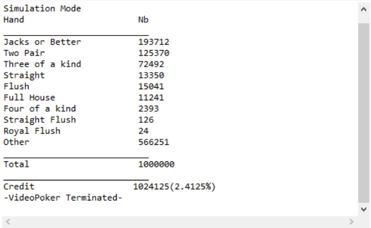
The theoretical return of the basic strategy for 100.2%. For sake of showing we present results for 1 million iterations in *Simulation Mode*. For the sake of showing the program yielded 0.411%, 0.806%, -1.3515%, 2.4125%, 0.552% in the following figures.



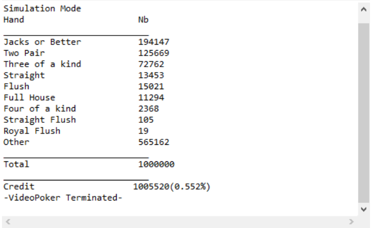
```
Simulation Mode
Hand                      Nb
____________________________
Jacks or Better           193722
Two Pair                  126145
Three of a kind           72849
Straight                  13459
Flush                     15039
Full House                11211
Four of a kind            2388
Straight Flush            101
Royal Flush               18
Other                     565068
____________________________
Total                     1000000
____________________________
Credit                   1004110(0.411%)
-VideoPoker Terminated-
```



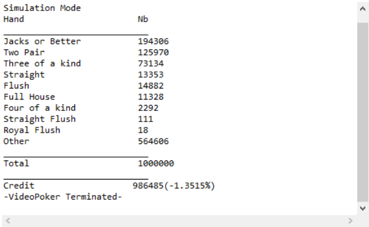
```
Simulation Mode
Hand                      Nb
____________________________
Jacks or Better           194051
Two Pair                  125691
Three of a kind           72987
Straight                  13537
Flush                     14762
Full House                11288
Four of a kind            2343
Straight Flush            87
Royal Flush               24
Other                     565230
____________________________
Total                     1000000
____________________________
Credit                   1008060(0.806%)
-VideoPoker Terminated-
```



```
Simulation Mode
Hand                      Nb
____________________________
Jacks or Better           193712
Two Pair                  125370
Three of a kind           72492
Straight                  13350
Flush                     15041
Full House                11241
Four of a kind            2393
Straight Flush            126
Royal Flush               24
Other                     566251
____________________________
Total                     1000000
____________________________
Credit                   1024125(2.4125%)
-VideoPoker Terminated-
```



```
Simulation Mode
Hand                      Nb
____________________________
Jacks or Better           194147
Two Pair                  125669
Three of a kind           72762
Straight                  13453
Flush                     15021
Full House                11294
Four of a kind            2368
Straight Flush            105
Royal Flush               19
Other                     565162
____________________________
Total                     1000000
____________________________
Credit                   1005520(0.552%)
-VideoPoker Terminated-
```



```
Simulation Mode
Hand                      Nb
____________________________
Jacks or Better           194306
Two Pair                  125970
Three of a kind           73134
Straight                  13353
Flush                     14882
Full House                11328
Four of a kind            2292
Straight Flush            111
Royal Flush               18
Other                     564606
____________________________
Total                     1000000
____________________________
Credit                   986485(-1.3515%)
-VideoPoker Terminated-
```

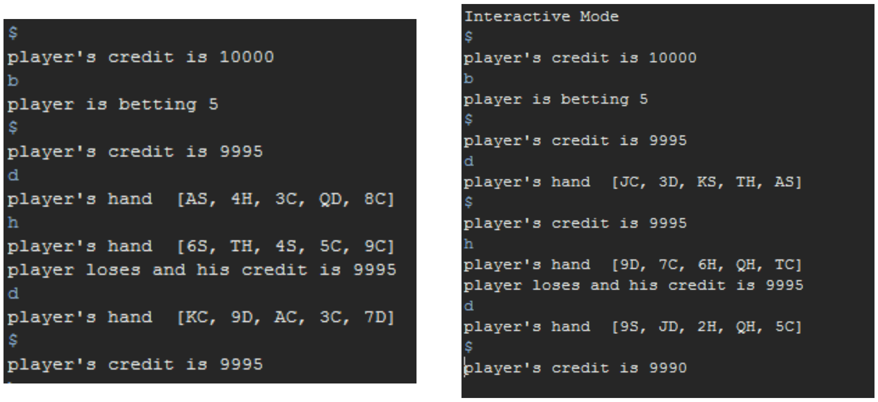
These results are an indicator of a good basic strategy despite a little variance on the outcome.

6 Details and nuances

Whenever a bet is made, value of the bet is subtracted to the player's credit. However, after the first bet, the player may choose to just *deal* and the bet must still be subtracted on the *deal* command and not only after the outcome of the play. This is illustrated in the figure 3.

Most of the variables were initialized at *null* not only because it is a good programming pratice so that these do not point toward "random" positions of memory but also for security purposes. Moreover, most of the variables of each class should be initialized as *private* and be accessed through the use of *getters and setters*. We did this in most of them.

Due to lack of time and the fact that we decided the stir our focus to major issues on the project, some prints, are printed with the generalized *toString* methods. For instance when in debug mode, the command **h 1 2** is printed to the terminal as **h [1,3]** for it is printed with the embbeded toString from ArrayList. This could be done by creating a class that extends from ArrayList and overriding *toString*, using *StringBuilder*, or hard coding it by printing only the specific part of the array.



```
$
player's credit is 10000
b
player is betting 5
$
player's credit is 9995
d
player's hand  [AS, 4H, 3C, QD, 8C]
h
player's hand  [6S, TH, 4S, 5C, 9C]
player loses and his credit is 9995
d
player's hand  [KC, 9D, AC, 3C, 7D]
$
player's credit is 9995
```

(a) Faulty credit situation.

```
Interactive Mode
$
player's credit is 10000
b
player is betting 5
$
player's credit is 9995
d
player's hand  [JC, 3D, KS, TH, AS]
$
player's credit is 9995
h
player's hand  [9D, 7C, 6H, QH, TC]
player loses and his credit is 9995
d
player's hand  [9S, JD, 2H, QH, 5C]
$
player's credit is 9990
```

(b) Non faulty credit situation.

Figure 3: Credit nuance.

7 Final remarks

Summing up, this project was not only highly educational but extremely interesting. The theme totally captivated us and motivated us to work on the project. The whole process of programming made us understand the value of Java and other object oriented languages. It is very easy to look at the project as a whole and work on its parts. Kind of a *divide to conquer* programming language. We misjudged the time that the Strategy would take which hindered its not so perfect results.