

Sevens

Motivation

Currently the game Sevens requires four people to play and a pack of cards. This program hopes to allow a person to: play Sevens on their own – the other players can be simulated by the program; play if they have no cards – as the cards are internally represented and can be displayed on screen when the player needs to see them; and also prevent cheating as the program stops the player being able to look at others cards or from playing an illegal move (passing when the player must play).

Description and Explanation

The most interesting aspects of the program will be the designing of algorithms that can consistently pick moves that are both legal and of strategic value. Core, but less interesting problems are how to display the state of play to the user and how to represent state of play internally.

Internal representation is important in that without the game cannot even begin to proceed. Initially the program solely used lists to store data and the data relevant to each player was then stored in separate elements for each player eg. [human com1 com2 com3]. Within each element of the first super list there would be four sub lists with each sublist referring to a separate suit. eg. [[[clubs][diamonds][hearts][spades]]

```
[...]
[...]
[...]
```

Each suit list would contain a set of integers [1 2 3] [] [] [] refers to the Ace, two and three of Clubs.

However, this form of storing the data was not readily understandable to other programmers. It also meant that as each sub list increased in size (containing data other than just the player's hand – eg. name, last move, etc...) it would take more operations to reach the elements at the end of the list – such as the 4th player's suit of spades. It was then decided to use objectclass to store the data. This would increase efficiency as accessing data for com3 would now be speeded up as the program would no longer have to scan through the entire list, but instead look inside a certain slot for each player object. In addition where the data was stored would now be much more accessible to a programmers – to retrieve a player's hand the call would now be “hand(player)” rather than “playerList(x)(y)”. It would also provide a way of setting the procedure used to get a move from a player as a unique property for each player, but a uniform way of accessing that specific procedure for that player by storing it in a slot. Also some data in the slots would no longer change during the game – such as a name slot. As such this slot could now also be used as an identifier for the object, for example – the name slot of a player object.

Displaying the state of play to the user is vitally important as the program needs to provide all the available to the user whilst not overloading the user with spurious information. In the end the decided output was as follows on the next page -

```

Rob played the Q of Hearts
Dan passed
Tom played the 6 of Hearts

Rob [10]  Dan [11]  Tom [9]
Clubs    [                ]
Diamonds [          6 7    ]
Hearts   [          6 7 8 9 10 J Q ]
Spades   [          5 6 7    ]
    Your hand is -
Clubs[2] Diamonds[4 5 10 K] Hearts[3 4 K] Spades[4 J]

```

The first three lines display the moves of the computer players since the user's last go. Then next line displays the number of cards each opponent has. The next four lines describe the board of the game with each line reserved for a unique suit. Within the square brackets are enough spaces for each of the 13 cards in the suit – these are filled in as each card is played. Finally, the last two lines describe the user's hand.

The number of lines of code in the program quickly exploded and it became hard to locate certain procedures or remember what they were for. As such the program was split up into for separate files: sevens; main; UI (user interface); and player. The two new main files were UI and main. UI contained all the code that resulted in output to the output screen – therefore any code found in UI either outputs code to the screen or is called by a procedure in UI to help produce output strings. Main on the other hand was the core of the program and applied all the transformations to the underlying data – such as: dealing, making a move, calculating legal moves and the main game procedures. Player held only the two procedures related to the player object, and sevens simply compiled the other files in the required order and called the main game procedure – sevens().

Finally we come to the AI procedures. First the algorithm that produced the move would need to consider which of its moves were legal and then which of these were the better moves.

- The first procedures simply randomly choose one of the legal moves.
- The second procedure produced a weight for each card depending upon its pip value. That is the two of hearts is considered a better move than the 6 of clubs because it is further away from the middle of the suit and opens up less possible moves for the other players – the 2 of hearts only allows the ace of hearts to be played, whereas the 6 of clubs would allow the five of clubs to be played which then allows the four of clubs to be played. This strategy reduces the ability of other players to play their own cards in the hope of forcing them to pass
- The final, and third, procedure also takes into account other cards in the player's hand. For example if, in addition to the two of hearts and 6 clubs, the player also held the four of clubs then the player would choose the 6 of clubs in the hope that another player would play the 5 of clubs, and thus enabling the player to play the four of clubs on a later go. This strategy lessens the probability of the player having to pass themselves. Unfortunately there wasn't enough time to implement this procedure as the current way of storing data (a player's legal moves), meant that the algorithm proved to be quite complex.

Usage Instructions

Starting

To start the program make sure all files are in the same folder and then open sevens.p and compile the whole file.

Using the program

The program will first ask you for a name. You may enter any non-empty value.

Then the program asks if you want to change the difficulty level. The larger the number, the better the computer algorithm

A new game will now begin and you will be asked to enter moves. To pass simply type 'pass'. To play card there are a number of options. To play the Jack of Hearts you may enter any of the following as the pip value: jack; j; or 11. For the suit you may enter either 'hearts' or 'h'. Cards should be entered as either 'pip of suit' or 'pip suit'. For example the jack of hearts may be entered as any of the following and more -

- Jack of Hearts
- jAck hEARTS
- 11 h
- 11 of h
- j h

Quitting

To quit press ctrl+c or decline to play another game when the current game ends.

Related work

help objectclass – good manual on objectclass works in pop11

help matches – contains information on how the pattern matcher is used in this program – especially with regards to using other procedures to modify inputs during matching or prevent matching taking place.

Limitations

The current limitations of the game are that it only allows one human player at a time. This is almost unavoidable in the fact that there is only one screen and it is therefore very hard to build the game in such a way as to not let players see their opponents hands. Another limitation is that some sources suggest that Sevens can be played with as little as 3 players and as many as 7, whilst the current game only allows for 4 players. One final limitation is that the game uses a text user interface which would certainly not make the game popular were it ever to get any where near a market of some kind.

Conclusions

Perhaps one of the more major stumbling blocs of this project was its sudden change in how it stored card data. For the majority of the time it uses a set of 4 sub lists in a list. However, when evaluation legal moves it switches to a list a sub lists where each sub list represents a single card i.e. [11 3] is the 'Jack of Hearts'. This was done to make analysis of whether a human had chosen a legal move. The program could simply transform 'Jack' into 11 and 'Hearts' into 3 and then check to see if the input was present in the list of legal moves. This later made evaluation of good moves for a computer player harder as each card is isolated with no reference to any other cards. Whereas, the list [2 4 10] (which represents the two, four and ten of a suit), allows an algorithm to easily see that playing the 4 is beneficial (it allows the three to be laid so that the two can be laid).

Due to how the program was built many of the limitations addressed earlier are easily solved. The limitation on the number of players is easily solved in that the procedures that call the output methods for the text interface, such as printBoard, could be called by humanMove, thus making the while loop more uniform and simply allowing it to cycle through lists rather than using separate ifs and elseifs to check whose turn is it and what to

do. The game is also split up into its core components and its user interface, allowing the text user interface to be easily stripped away and a GUI built and plugged in instead. Making the game multi-player would, however, require knowledge of how to use networks with pop11 (if that is indeed possible) and thus makes a multi-player mode beyond the scope of this project.

Bibliography

<http://web.archive.org/web/20000519233854/www.terindell.com/asylum/games/fantan.html>

This website contains the rules of the main game plus those of many variants. In addition it also has some basic strategy tips.

<http://www.fwend.com/sevens.htm>

This is a Java implementation of Sevens but is unusual in that it has several variations

- the seven of hearts must be played first;
- the eight must be played as the second card of each suit, before the six;
- as soon as one player runs out of cards, penalty points are scored by all the other players

Appendix A – complete code tests

For unit testing please see procedure headers in Appendix B.

Complete code tests -

Below tests to see if all procedures work correctly together. It contains to completed games to see if the game correctly remembers game stats from one game to the next and if to see if the option to play another game or not works as intended.

```

What is your name?
?
? Jack E Hargreaves
Do you want to adjust the difficulty levels for the computer players? y/n
? y
Please pick a level between 1 and 2.
Set the difficulty level for Rob.
? 1
Set the difficulty level for Dan.
? 2
Set the difficulty level for Tom.
? 2
** Dan to start
Dan played the 7 of Hearts
Tom played the 7 of Clubs

Rob [13]  Dan [12]  Tom [12]
Clubs   [           7           ]
Diamonds [           ]
Hearts  [           7           ]
Spades  [           ]

Your hand is -
Clubs[5 9 Q] Diamonds[4 7 10 Q] Hearts[3 5 6 Q] Spades[3 Q]
? pass
That is not a legal move.
? 8 d
That is not a legal move.
? 4 d
That is not a legal move.
? 6 z
Move not recognised.
? whoops
Move not recognised.
? 1 c
Move not recognised.
? 7 d

Rob played the 6 of Diamonds

```

Dan played the 8 of Clubs
Tom played the 5 of Diamonds

```
Rob [12] Dan [11] Tom [11]
Clubs   [           7 8           ]
Diamonds [           5 6 7         ]
Hearts  [           7             ]
Spades  [                   ]
```

Your hand is -

Clubs[5 9 Q] Diamonds[4 10 Q] Hearts[3 5 6 Q] Spades[3 Q]
? 9c

Rob played the 8 of Diamonds
Dan played the 8 of Hearts
Tom played the 10 of Clubs

```
Rob [11] Dan [10] Tom [10]
Clubs   [           7 8 9 10       ]
Diamonds [           5 6 7 8       ]
Hearts  [           7 8           ]
Spades  [                   ]
```

Your hand is -

Clubs[5 Q] Diamonds[4 10 Q] Hearts[3 5 6 Q] Spades[3 Q]
? 8h
That is not a legal move.
? 6h

Rob played the 9 of Diamonds
Dan played the 7 of Spades
Tom played the J of Clubs

```
Rob [10] Dan [9] Tom [9]
Clubs   [           7 8 9 10 J     ]
Diamonds [           5 6 7 8 9     ]
Hearts  [           6 7 8         ]
Spades  [           7             ]
```

Your hand is -

Clubs[5 Q] Diamonds[4 10 Q] Hearts[3 5 Q] Spades[3 Q]
? q c

Rob passed
Dan played the 6 of Spades
Tom played the K of Clubs

```
Rob [10] Dan [8] Tom [8]
Clubs   [           7 8 9 10 J Q K ]
Diamonds [           5 6 7 8 9     ]
Hearts  [           6 7 8         ]
Spades  [           6 7           ]
```

Your hand is -

Clubs[5] Diamonds[4 10 Q] Hearts[3 5 Q] Spades[3 Q]
? 10 d

Rob passed
Dan played the 8 of Spades
Tom played the J of Diamonds

```
Rob [10] Dan [7] Tom [7]
Clubs   [           7 8 9 10 J Q K ]
Diamonds [           5 6 7 8 9 10 J ]
Hearts  [           6 7 8         ]
Spades  [           6 7 8         ]
```

Your hand is -

Clubs[5] Diamonds[4 Q] Hearts[3 5 Q] Spades[3 Q]
? q d

Rob played the 9 of Spades
Dan played the K of Diamonds
Tom played the 9 of Hearts

```
Rob [9] Dan [6] Tom [6]
Clubs   [           7 8 9 10 J Q K ]
Diamonds [           5 6 7 8 9 10 J Q K ]
Hearts  [           6 7 8 9         ]
Spades  [           6 7 8 9         ]
```

Your hand is -

Clubs[5] Diamonds[4] Hearts[3 5 Q] Spades[3 Q]
? 5h

Rob played the 10 of Spades
 Dan played the 10 of Hearts
 Tom played the J of Spades

Rob [8] Dan [5] Tom [5]
 Clubs [7 8 9 10 J Q K]
 Diamonds [5 6 7 8 9 10 J Q K]
 Hearts [5 6 7 8 9 10]
 Spades [6 7 8 9 10 J]
 Your hand is -
 Clubs[5] Diamonds[4] Hearts[3 Q] Spades[3 Q]
 ? q s

Rob played the K of Spades
 Dan played the J of Hearts
 Tom played the 5 of Spades

Rob [7] Dan [4] Tom [4]
 Clubs [7 8 9 10 J Q K]
 Diamonds [5 6 7 8 9 10 J Q K]
 Hearts [5 6 7 8 9 10 J]
 Spades [5 6 7 8 9 10 J Q K]
 Your hand is -
 Clubs[5] Diamonds[4] Hearts[3 Q] Spades[3]
 ? q h

Rob played the 4 of Spades
 Dan played the K of Hearts
 Tom played the 6 of Clubs

Rob [6] Dan [3] Tom [3]
 Clubs [6 7 8 9 10 J Q K]
 Diamonds [5 6 7 8 9 10 J Q K]
 Hearts [5 6 7 8 9 10 J Q K]
 Spades [4 5 6 7 8 9 10 J Q K]
 Your hand is -
 Clubs[5] Diamonds[4] Hearts[3] Spades[3]
 ? 5c

Rob passed
 Dan played the 4 of Hearts
 Tom passed

Rob [6] Dan [2] Tom [3]
 Clubs [5 6 7 8 9 10 J Q K]
 Diamonds [5 6 7 8 9 10 J Q K]
 Hearts [4 5 6 7 8 9 10 J Q K]
 Spades [4 5 6 7 8 9 10 J Q K]
 Your hand is -
 Diamonds[4] Hearts[3] Spades[3]
 ? 4d

Rob played the 3 of Diamonds
 Dan played the 4 of Clubs
 Tom played the 2 of Diamonds

Rob [5] Dan [1] Tom [2]
 Clubs [4 5 6 7 8 9 10 J Q K]
 Diamonds [2 3 4 5 6 7 8 9 10 J Q K]
 Hearts [4 5 6 7 8 9 10 J Q K]
 Spades [4 5 6 7 8 9 10 J Q K]
 Your hand is -
 Hearts[3] Spades[3]
 ? 3h

Rob played the 2 of Hearts
 Dan passed
 Tom played the 3 of Clubs

Rob [4] Dan [1] Tom [1]
 Clubs [3 4 5 6 7 8 9 10 J Q K]
 Diamonds [2 3 4 5 6 7 8 9 10 J Q K]
 Hearts [2 3 4 5 6 7 8 9 10 J Q K]
 Spades [4 5 6 7 8 9 10 J Q K]
 Your hand is -
 Spades[3]
 ? 3s

** win state is **

```

Rob [4] Dan [1] Tom [1]
Clubs [ 3 4 5 6 7 8 9 10 J Q K]
Diamonds [ 2 3 4 5 6 7 8 9 10 J Q K]
Hearts [ 2 3 4 5 6 7 8 9 10 J Q K]
Spades [ 3 4 5 6 7 8 9 10 J Q K]

```

you're the winner

```

Rob's hand is - Clubs[2] Diamonds[A] Hearts[A] Spades[A]
Dan's hand is - Clubs[A]
Tom's hand is - Spades[2]

```

GAMES WON

Jack E Hargreaves: 1

Rob: 0

Dan: 0

Tom: 0

Total games played: 1

Do you want to play again? yes/no

? y

** Tom to start

Tom passed

```

Rob [13] Dan [13] Tom [13]

```

```

Clubs [ ]

```

```

Diamonds [ ]

```

```

Hearts [ ]

```

```

Spades [ ]

```

Your hand is -

```

Clubs[3 4 6] Diamonds[A Q] Hearts[6 8 K] Spades[3 6 7 J Q]

```

? 7s

Rob played the 7 of Hearts

Dan passed

Tom passed

```

Rob [12] Dan [13] Tom [13]

```

```

Clubs [ ]

```

```

Diamonds [ ]

```

```

Hearts [ 7 ]

```

```

Spades [ 7 ]

```

Your hand is -

```

Clubs[3 4 6] Diamonds[A Q] Hearts[6 8 K] Spades[3 6 J Q]

```

? 8h

Rob played the 7 of Clubs

Dan played the 9 of Hearts

Tom played the 8 of Clubs

```

Rob [11] Dan [12] Tom [12]

```

```

Clubs [ 7 8 ]

```

```

Diamonds [ ]

```

```

Hearts [ 7 8 9 ]

```

```

Spades [ 7 ]

```

Your hand is -

```

Clubs[3 4 6] Diamonds[A Q] Hearts[6 K] Spades[3 6 J Q]

```

? 6c

Rob played the 7 of Diamonds

Dan played the 9 of Clubs

Tom played the 10 of Clubs

```

Rob [10] Dan [11] Tom [11]

```

```

Clubs [ 6 7 8 9 10 ]

```

```

Diamonds [ 7 ]

```

```

Hearts [ 7 8 9 ]

```

```

Spades [ 7 ]

```

Your hand is -

```

Clubs[3 4] Diamonds[A Q] Hearts[6 K] Spades[3 6 J Q]

```

? 6s

Rob played the 8 of Diamonds

Dan played the J of Clubs

Tom played the 5 of Clubs

```

Rob [9] Dan [10] Tom [10]

```

```

Clubs [ 5 6 7 8 9 10 J ]

```

```

Diamonds [ 7 8 ]

```

Hearts [7 8 9]
 Spades [6 7]
 Your hand is -
 Clubs[3 4] Diamonds[A Q] Hearts[6 K] Spades[3 J Q]
 ? 4c

Rob played the 10 of Hearts
 Dan played the J of Hearts
 Tom played the 9 of Diamonds

Rob [8] Dan [9] Tom [9]
 Clubs [4 5 6 7 8 9 10 J]
 Diamonds [7 8 9]
 Hearts [7 8 9 10 J]
 Spades [6 7]
 Your hand is -
 Clubs[3] Diamonds[A Q] Hearts[6 K] Spades[3 J Q]
 ? 3c

Rob played the Q of Clubs
 Dan played the 5 of Spades
 Tom played the K of Clubs

Rob [7] Dan [8] Tom [8]
 Clubs [3 4 5 6 7 8 9 10 J Q K]
 Diamonds [7 8 9]
 Hearts [7 8 9 10 J]
 Spades [5 6 7]
 Your hand is -
 Diamonds[A Q] Hearts[6 K] Spades[3 J Q]
 ? 6h

Rob played the 8 of Spades
 Dan played the 4 of Spades
 Tom played the 2 of Clubs

Rob [6] Dan [7] Tom [7]
 Clubs [2 3 4 5 6 7 8 9 10 J Q K]
 Diamonds [7 8 9]
 Hearts [6 7 8 9 10 J]
 Spades [4 5 6 7 8]
 Your hand is -
 Diamonds[A Q] Hearts[K] Spades[3 J Q]
 ? pass
 That is not a legal move.
 ? 3s

Rob played the 5 of Hearts
 Dan played the 2 of Spades
 Tom played the A of Spades

Rob [5] Dan [6] Tom [6]
 Clubs [2 3 4 5 6 7 8 9 10 J Q K]
 Diamonds [7 8 9]
 Hearts [5 6 7 8 9 10 J]
 Spades [A 2 3 4 5 6 7 8]
 Your hand is -
 Diamonds[A Q] Hearts[K] Spades[J Q]
 ? pass

Rob played the Q of Hearts
 Dan passed
 Tom played the 4 of Hearts

Rob [4] Dan [6] Tom [5]
 Clubs [2 3 4 5 6 7 8 9 10 J Q K]
 Diamonds [7 8 9]
 Hearts [4 5 6 7 8 9 10 J Q]
 Spades [A 2 3 4 5 6 7 8]
 Your hand is -
 Diamonds[A Q] Hearts[K] Spades[J Q]
 ? k h

Rob played the A of Clubs
 Dan played the 3 of Hearts
 Tom played the 10 of Diamonds

Rob [3] Dan [5] Tom [4]
 Clubs [A 2 3 4 5 6 7 8 9 10 J Q K]

```

Diamonds [          7 8 9 10      ]
Hearts  [    3 4 5 6 7 8 9 10 J Q K]
Spades  [A 2 3 4 5 6 7 8          ]
Your hand is -
Diamonds[A Q] Spades[J Q]
? pass

```

```

Rob played the 2 of Hearts
Dan played the J of Diamonds
Tom played the A of Hearts

```

```

Rob [2] Dan [4] Tom [3]
Clubs   [A 2 3 4 5 6 7 8 9 10 J Q K]
Diamonds [          7 8 9 10 J      ]
Hearts  [A 2 3 4 5 6 7 8 9 10 J Q K]
Spades  [A 2 3 4 5 6 7 8          ]
Your hand is -
Diamonds[A Q] Spades[J Q]
? q d

```

```

Rob played the K of Diamonds
Dan passed
Tom played the 9 of Spades

```

```

Rob [1] Dan [4] Tom [2]
Clubs   [A 2 3 4 5 6 7 8 9 10 J Q K]
Diamonds [          7 8 9 10 J Q K]
Hearts  [A 2 3 4 5 6 7 8 9 10 J Q K]
Spades  [A 2 3 4 5 6 7 8 9        ]
Your hand is -
Diamonds[A] Spades[J Q]
? pass

```

```

Rob passed
Dan passed
Tom played the 10 of Spades

```

```

Rob [1] Dan [4] Tom [1]
Clubs   [A 2 3 4 5 6 7 8 9 10 J Q K]
Diamonds [          7 8 9 10 J Q K]
Hearts  [A 2 3 4 5 6 7 8 9 10 J Q K]
Spades  [A 2 3 4 5 6 7 8 9 10     ]
Your hand is -
Diamonds[A] Spades[J Q]
? j s

```

```

Rob passed
Dan passed
Tom played the 6 of Diamonds

```

** win state is **

```

Rob [1] Dan [4] Tom [0]
Clubs   [A 2 3 4 5 6 7 8 9 10 J Q K]
Diamonds [          6 7 8 9 10 J Q K]
Hearts  [A 2 3 4 5 6 7 8 9 10 J Q K]
Spades  [A 2 3 4 5 6 7 8 9 10 J   ]

```

```

your hand is - Diamonds[A] Spades[Q]
Rob's hand is - Diamonds[2]
Dan's hand is - Diamonds[3 4 5] Spades[K]
Tom is the winner

```

GAMES WON

Jack E Hargreaves: 1

Rob: 0

Dan: 0

Tom: 1

Total games played: 2

Do you want to play again? yes/no
? n
