Classic Blackjack Card Game in Java with 52 normal playing cards

By Tasmita Singh, Daanyaal Salie, and Vishma Ambelal Rama.

The Background and Motivation:

*Blackjack* is a slightly complex card game that is played with a full deck of 52 playing cards, and the suits don’t matter (Blackjack.org, 2019). A player plays against the dealer (can be the computer). The player makes his/her wager, and the dealer is asked to deal 2 cards to each player. To win you need to beat the dealer by either drawing a hand value that is higher than the dealer’s hand value, or by the dealer drawing a hand value that goes over 21, or even by drawing a hand value of 21 on your first two cards, when the dealer does not. This is called a “blackjack”. However, you will lose if your hand value exceeds 21, or the dealer’s hand has a greater value than yours at the end of the round and the dealer will win and you will have to forfeit your bet.

The card values are such that cards 2-10 are worth the value of the number on the face of the card. Face cards (those with pictures on them) are worth 10, except for the Ace, which is worth either 1 or 11. A picture combined with an Ace is called a “Blackjack” (a value of 21). After receiving your first two cards, the player has two initial options, either a hit or a stand. A hit is when the player will request another card from the dealer and if the third card is larger than 21 the player will forfeit this round and will be “Bust!”. The other option is a stand, now the player will decide whether they have more changes or receiving a higher card than a lower card, and so they don’t take another card.

Another option is called a double down, and its allowed straight after the first two cards have been dealt and the player is allowed to place an additional bet to equal the original wager but cannot ask for any more hits after this third card. The next option is called a split, such that if the players first two cards hold the same value (2 two’s or 2 eight’s) then the player can spilt them into two separate playing hands, the split hand becomes two separate bets and the dealer will hit you with a further card for each playing hands. After this split you are not allowed to use a hit, double-down or re-split them. The player can also choose to surrender if the dealers up-card (the card they show) is either an ace or a 10 value (remember this includes the picture cards) and will have to surrender half his bet to the dealer and a new round will begin. Insurance is when the dealers up-card is an ace, the player can take insurance against the chance that the dealer has a blackjack. If the dealer and the player both have the same total, it’s called a push and no money changes hands. The pay-outs are such that if the player beats the dealer with a higher combination the player receives a 1:1 pay-out (if you bet $10, you get $10), and if the player hits a “Blackjack” (10 value card plus an ace) the player receives a 3:2 pay-out(if you bet $10, you get $15).

A Clear Problem Statement:

The aim of this version of *Blackjack* for juniorsis to beat the dealer and get a blackjack after the first 2 cards have been handed, else the player tries to choose the best option mentioned above to get closest to 21 without exceeding 21. This game has been simplified for children to play, with lesser functions (than those seen in classic blackjack) and no betting taking place.

The Approach to solve the problem:

Having come up with an ideal game to code, it was easier to set milestones so our group members were motivated in the short-term. This project aims to implement a card game in Java with the following milestones: The first is to code a data structure a list of cards that can contain a standard deck of 52 cards. From 1-7 players can play this game at a time including computer player(s). The program works in such a way that according to the number of players in the game, it will call that many lists for each of the players hands. Each hand will be tracked by showing their current score and the cards that they hold. Each player will be given the option to choose a function they wish to perform, and then see what their new score and new hand is.

Finally following the rules of winning and losing the game will show the players final state in the game.

Finally, this work will be presented for assessment in the form of a GitHub code and Project Document, which includes the revision and extension of this proposal. The extensions will include a walkthrough of the steps through the SDLC that our group followed, and a Visual Guide that shows the game in action (almost like a demo). One way to do this is in the form of ‘storyboards’ with an explanation per frame, that at least shows the start of the game, playing example of the game, and end of the game.

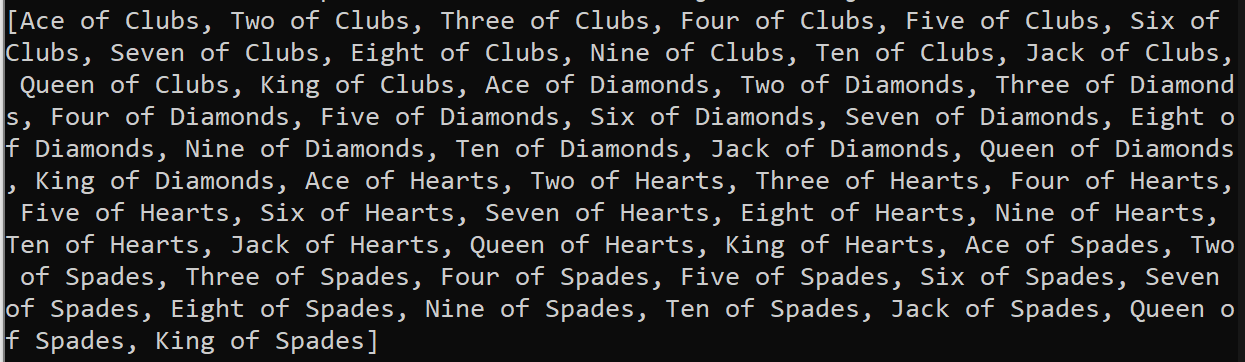
Walk-through the SDLC approach to the Blackjack Card Game

An SDLC is a framework that describes the order in which activities

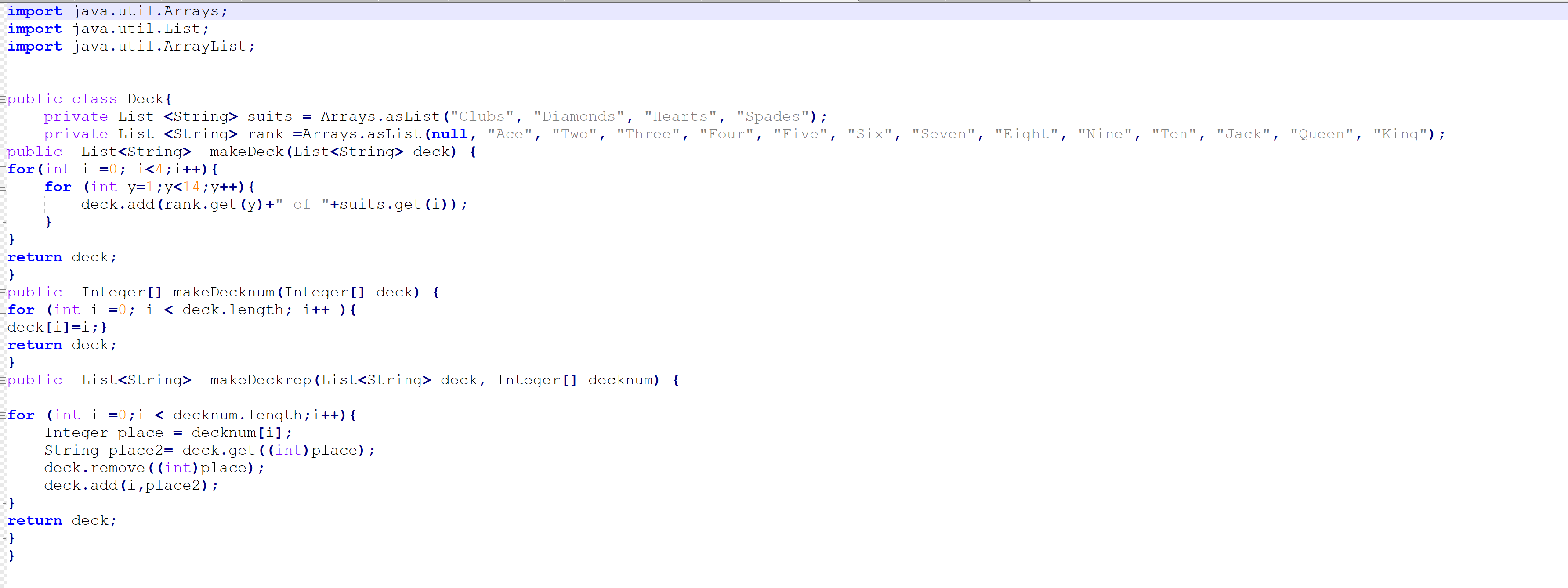
are performed at each stage of a software-based project. Agile project management is an iterative and incremental approach to delivering requirements throughout the project life cycle. At the core, agile projects should exhibit central values and behaviours of trust, flexibility, empowerment and collaboration. Due to the flexibility we chose this approach and we could afford to meet regularly (once every week for 3-4 weeks and then every second day for the last week). During our meeting we coded together and sorted out errors together rather than split the work which may have created problems when creating the ‘*Blackjackmain’* class.

Visual Guide that shows the game in action

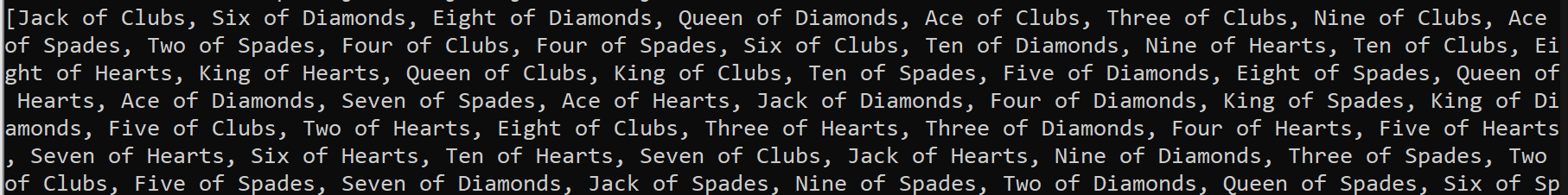
UNSHUFFLED CARDS



USING A ‘RANKS LIST’ AND A ‘SUITS LIST’ TO EASILY CREATE ALL 52 CARDS WITHOUT HAVING TO CODE FOR EACH OF THE 52 CARDS.



SHUFFLED CARDS (ASSIGNING A RANDOM INDEX NUMBER WITHIN THE RANGE OF 1-52 TO A STRING INDEX OF THE CARDS NAME THAT ARE NOT SHUFFLED YET (I.E. INDEX 3 FROM THE RANDOM NUMBERS FUCTION AND FROM THE CARDS BRINGS A “THREE OF SPADES”)

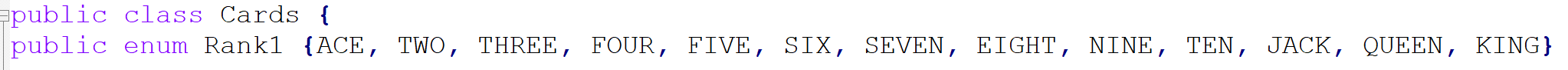


EACH PLAYER GETTING THEIR FIRST HAND OF CARDS

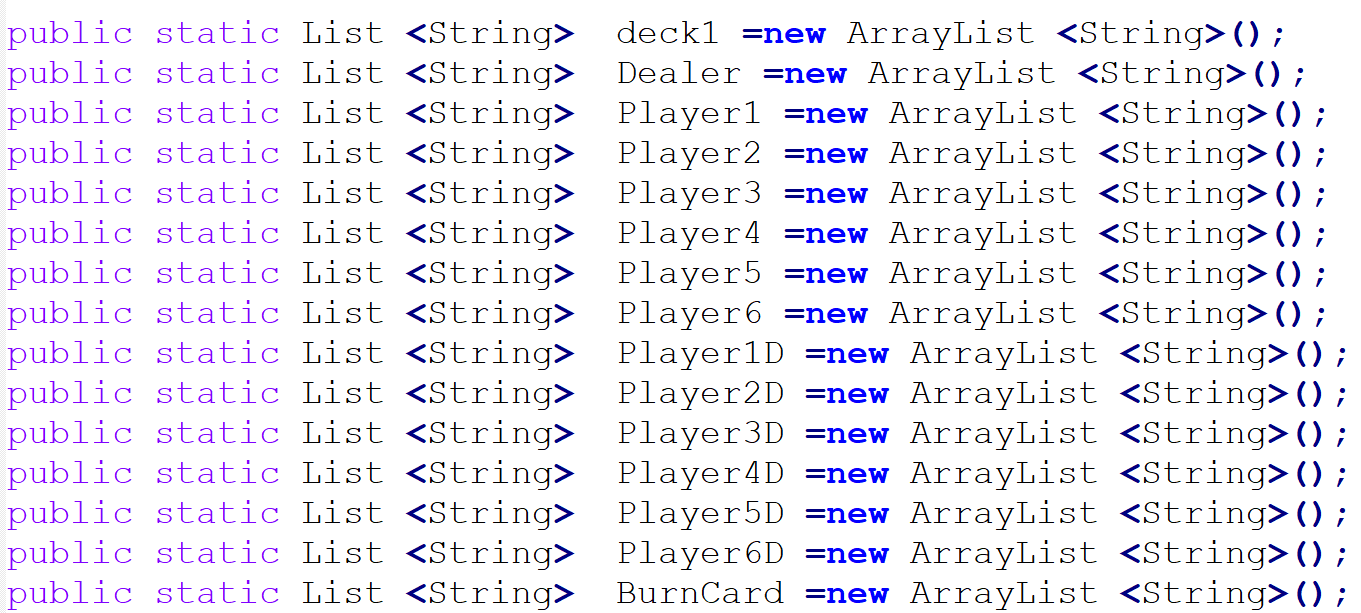
SHANNON ENTROPY WORKING (SHOWING THE LEVEL OF SHUFFLED-NESS, WHERE THE BEST POSSIBLE SOLUTION IS 5.7)



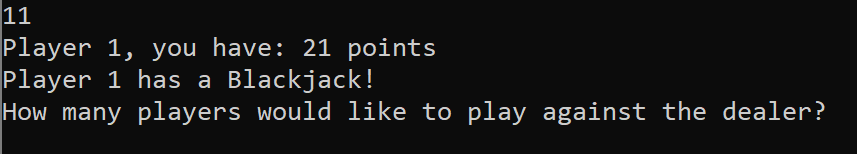
SHOWING THE ENUMS WORKING (ASSIGNING VALUES TO EACH CARD), WHERE THE NUMBER CARDS ARE THEIR FACE VALUES AND THEN THE PICTURE CARDS ARE 10, AND THE ACE IS A SPECIAL CARD THAT’S WORTH EITHER A 1 OR 11.



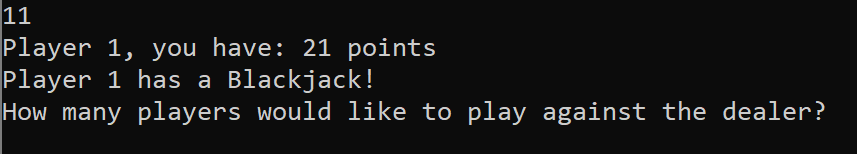
CREATING 7 LISTS FOR ALL THE PLAYERS AND ASKING HOW MANY PLAYERS WILL PLAY THE GAME AND THEN CALL THAT MANY LISTS. FINALLY DEPENDING ON THE NUMBER OF PLAYERS THEY WILL EACH BE DEALT 2 CARDS FROM THE SHUFFLED DECK. THE CARDS THAT HAVE BEEN DEALT WILL BE REMOVED FROM THE ORIGINAL SHUFFLED DECK OF CARDS.



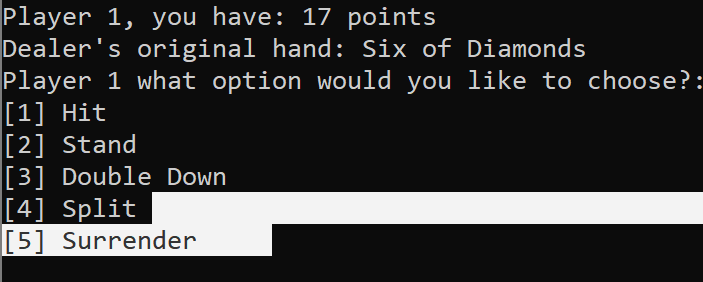
THE PLAYERS ORIGINAL SCORE WILL BE DISPLAYED, AND THEN THE PLAYERS EACH GET AN OPTION THAT WILL CALL A FUNCTION TO DO CERTAIN ACTIONS, DESCRIBED IN THE PROPOSAL.



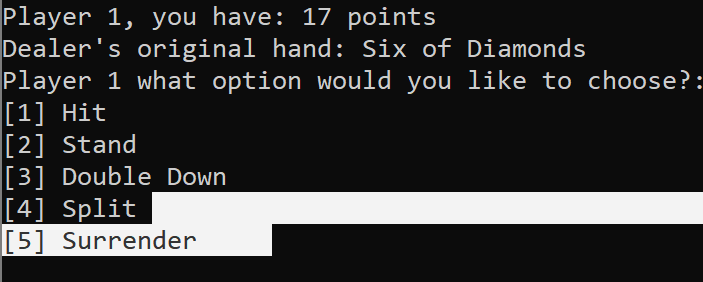
IF THE PLAYER 1 GETS 21 ON THEIR FIRST HAND, IT’S CALLED A “BLACKJACK”



SINCE PLAYER 1 DOESN’T HAVE 21 ON THEIR FIRST HAND, PLAYER 1, FOR EXAMPLE WILL CHOOSE TO TAKE A HIT, AND BE GIVEN ANOTHER CARD. THEIR SCORE WILL ALSO BE UPDATED.



IF THE PLAYER 1 EXCEEDS 21, THEY ARE “BUST” AND THEY LOSE THEIR ORIGINAL BET.



The Gantt Table with the milestones:

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Start(weeks)** | **Finish(weeks)** |
| Choosing the card game | 0 | 0.2 |
| Breaking down the different functions needed for the game to work | 0 | 0.3 |
| Creating the deck of cards(unshuffled) | 1 | 2 |
| Shuffle the cards (using Fisher-Yates) | 2 | 3 |
| Test the level of shuffling (Shannon entropy) | 2 | 3 |
| Create the cards class with the Enums to assign values to each card | 3 | 3.2 |
| Create the different hands for each player (2-7 players) | 3.2 | 4 |
| Create a blackjack functions class with different options | 4 | 4.6 |
| Implement the blackjack main function to call all the other functions | 4.6 | 4.6 |
| Test the blackjack main | 4.7 | 5 |

Gantt Chart:

List of References:

J. Mayers. Blackjackapprenticeship.com. (2019). [online] Available at: https://www.blackjackapprenticeship.com/how-to-play-blackjack/ [Accessed 1 Sep. 2019].

Blackjack.org. (2019). *Blackjack Rules - Learn How to Play 21 - [Tips & Best Practices]*. [online] Available at: https://www.blackjack.org/blackjack-rules/ [Accessed 1 Sep. 2019].

Mjlawler (2017). *Card Shuffling and Shannon Entropy*. [online] Mike's Math Page. Available at: https://mikesmathpage.wordpress.com/2017/04/23/card-shuffling-and-shannon-entropy/ [Accessed 1 Sep. 2019].