Problem Definition:

Blackjack is arguably one of the most played card games in the world, due to its “simple to learn but difficult to master” nature, as it is characterised by luck mainly however skill does play a part in whether the player wins or loses. Blackjack is well known for its very low house edge on the player therefore it is often played by both skilled and casual gamblers for their chance to win prizes. However, as the world progresses there is a shift to create a digital version of many physical card games, and the problem proposed is to create a well functioning digital version of Blackjack. This version of blackjack is also to include a plethora of features, such as truly randomised, endless decks, automatic scoring and betting, customizable AI and computer players to play with.

My solution to this problem is creating a Javascript application featuring the JQuery language as it contains a variety of pre-existing libraries making it easier to create a strong object oriented and modular solution. Javascript was also chosen due to the new surge in popularity of web applications.

Needs and Objectives:

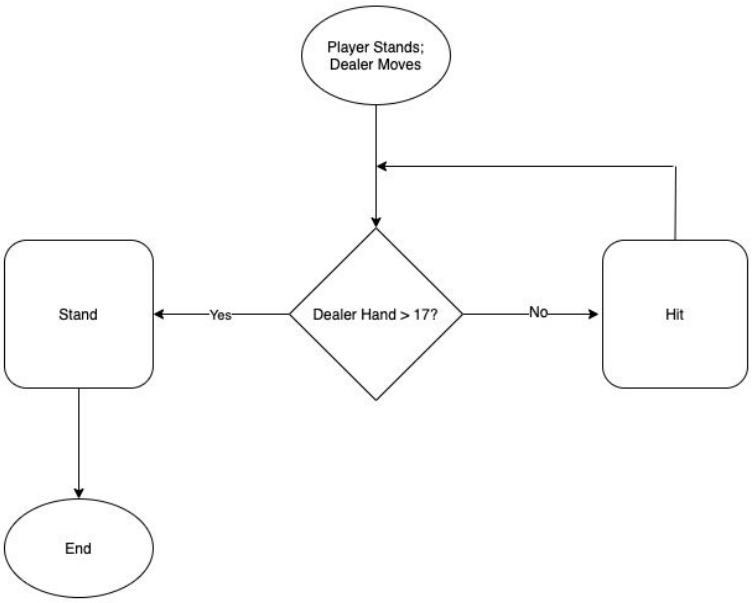
Needs:

- Modular code base which is object orientated.
- Decks that can be shuffled
- An ergonomic UI design and user experience.
- Computer opponents that can be customised
- A form of a betting system
- An automatic dealer that can be customized as well

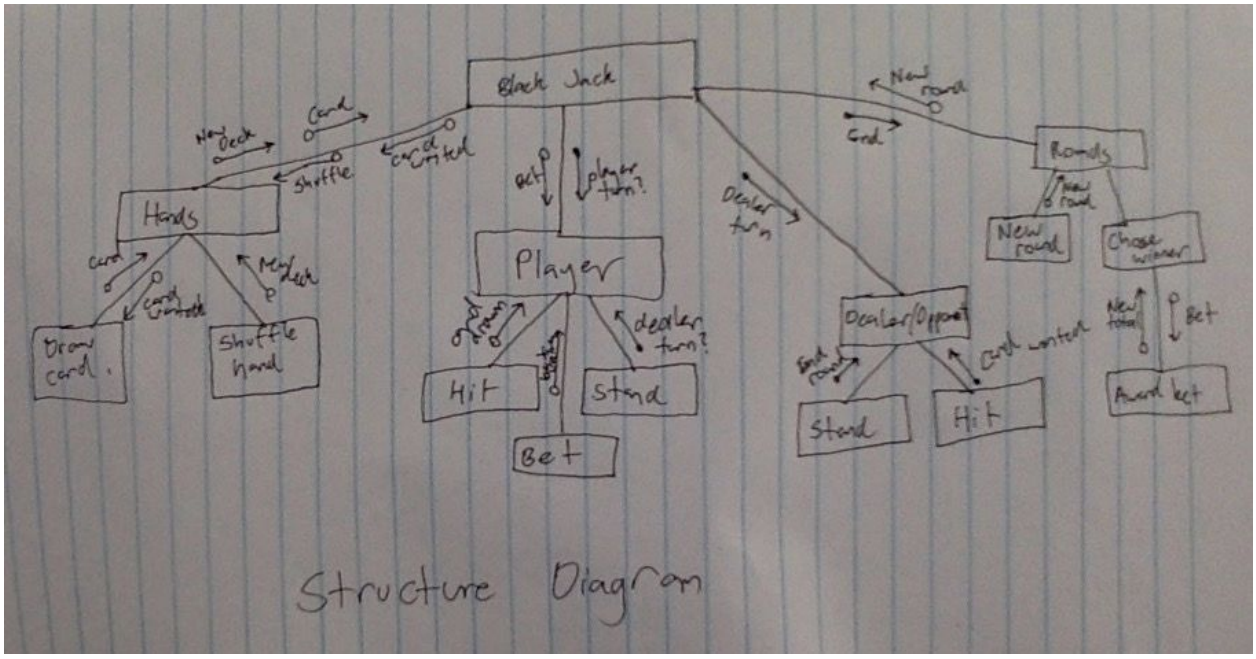
Objectives:

The objective of this program is to provide the end user with a fun, simplistic, entertaining version of Blackjack that fully emulates the physical card game as well as providing new features such as customisable AI players and an automatic dealer.

Dealer Flowchart:



Structure Diagram:



Evaluation of Implementation Methods:

Direct Cutover method:

A direct cutover is where the old system is immediately moved to the new system, causing the old system obsolete once the changes are made. Advantages of this method is that it is quick

and inexpensive, however this method is stressful for users as well as high risk as bugs that were overlooked during the development stages can be found with no way of going back to the original solution.

Parallel Implementation Method:

Parallel implementation is where the old and new system is active where users are allowed to become familiar with the new system. This creates less stress for developers and as they are able to fix newly discovered bugs in the new system and roll back the system if something goes disastrous. Advantages include allowing users to transition at their own will and making it easier to fix bugs in case they are found. The disadvantages of this method are that some users will stay on the old system and there will create a division in the user base, as well as the fact that it is more costly to run two systems simultaneously, and the data being created on the systems must be synced together somehow.

Pilot implementation:

Pilot implementation is characterised by a new system being introduced to a small portion of a user group for trial. A phased approach has many advantages, one being the project timeline. Often a project is rushed due to timeline constraints and the players involved simply charge forward to get the project completed. This will result in missing requirements as well as a lack of proper testing and tweaking - which results in a lackluster outcome. Using phases can help avoid this wasted effort. If time is spent qualifying requirements up-front, everyone involved will be able to plan and schedule the project efficiently. Pilot implementation however takes a long time to implement the new system therefore the difficulty of development increases as well as the cost of having to manage two systems.

Phased implementation:

A phased implementation is a combination of both direct and parallel implementation methods. In this method the old system is slowly phased out and replaced with new matching features from the new system, therefore slowly implementing the new features to the user base. Advantages are that the slow rate of change allows users to become more accustomed with the new software, and forces users not willing to switch systems. Disadvantages are that it takes a long time to implement the new system, and that the difficulty of development that occurs when having two different systems. It also comes at a cost of managing two systems.

Verdict:

For my system I will be using the parallel implementation method as I am hosting my code on GitHub so therefore I can freely change between different branches of my repository therefore allowing me to change between old and development versions of my system. As I am the only developer on the project, there is no need to use more complicated methods of implementation.