



MONASH  
University

## ASSIGNMENT 2

Programming Paradigms

Haider Ali

Student ID: 31371337

**FIT2102**

Due Date: 27th October, 2021

## **Introduction:**

This assignment challenged me to demonstrate my understanding of functional programming paradigms and to learn a new language, Haskell. I had a bit of issues grasping the syntax used in Haskell but after six weeks, I have managed to make a lot of progress. Sadly, it was the end of the semester by the time I had really managed to understand how Haskell works and it was actually fun to code in.

For this assignment, I had to write a Heuristic AI that selected what action a player should perform given a hand of cards. playCard function took in the dealers up-card and player information to determine the best action that the player should perform to get maximum points.

## **PlayCard:**

I have not played blackjack before so I had to research online to find how the game worked to come up with a strategy that would give the player a considerable chance to beat the dealer. I have come up with a Heuristic AI / I use pattern matching in my playCard function to get the dealers up card information and the player's current hand. Based upon these two factors, the player has to Hit or Stand (in my implementation). Sadly, I was not able to make other actions such as DoubleDown, Split and Insurance to work properly on time and I was forced to remove them from my implementation. My makeBid function finds the points the current player has and makes a bid according to the current players hand. If the player has a combo then we need to bid the maximum amount otherwise we can bid the minimum amount.

## **Other Haskell features/ Parser Combinators:**

Using pattern matching to handle Maybes in case of memory since the memory is Nothing(blank) at the start. Using Maybes in case of the dealers up card to check if it is the first round of the game. The players need to bid and checking for the dealer card being Nothing means it's the first round.

Using parsers to handle the memory as the memory is used for storing the players bid, the players id and the players action. I have copied over and reused the parsers we were given in our week 11 tutorial since they help me parse in strings, characters and even something between specified character / string using the betweenStringTok/betweenCharTok. My Actions are between a '(' and a ':' so I use betweenCharTok to extract actions. My Bids are between a "(Bid:" and a ":" so I use betweenStringTok to get bids from the memory. I am using the getResult function to get the values (in string) from the ParseResult we get after we parse. The getActionMem function is designed so that we can check if the player has made any other action rather than bid. It will helps us to keep track if we can doubledown. We retrieve the players bid using the getPlayerBidMem function. We need to parse the bid amount in memory and use that in DoubleDown (double down removed).

## **BNF:**

<bid> = "100" | "10"

<action> = "Hit" | "Stand"

<player\_id> = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<mem> = "("(<action>":"<player\_id>";")("Bid:"<bid>":"<player\_id>";")<mem>+

## Conclusion:

I had fun with Haskell in the last few days. Sadly, I wasn't able to quickly grasp the Haskell syntax and it lead me to wasting more time learning rather than getting started with the assignment as quickly as I could. I learned a bit about parser combinators and how important they are in various other aspects such as using them in JSON and other forms of data handling. I enjoyed seeing my player move up and down the rank ladder and am proud of it getting atleast a rank that is around the 50% even though it has a Heuristic AI and manages to defeat two bots.

Welcome halio010

Upload a new player [here](#).

View upload [status](#).

Your current results are:

Id	Rank	Elo
656	83	1248.3953

View on the [ladder](#).

Download logs [here](#).

You can click on the number of hands played to see the game.

Time	Score	Hands played
Oct 27 17:16:16	10235	100
Oct 27 17:16:13	10025	100
Oct 27 16:29:36	9995	100
Oct 27 16:25:01	9870	100

references:

[http://www.blackjackbee.com/strategy/hit\\_or\\_stand.php](http://www.blackjackbee.com/strategy/hit_or_stand.php)

<https://www.beatblackjack.org/en/strategy/>

<https://tgdwyer.github.io/parsercombinators/>