Correction Notes - Assignment 5

Group: Manzil, Cüneyt and Dhananjay

Task 1

I am really confused by the offset in the player's score. You must have thought about this and selected this deliberately, but I don't see why. The dealers policy and the reward function are absolutely correct. Your interpretation of the player's 'hit' action is not right. When 'hit' is selected, the player will draw *one* card, and then decide again if the state is not terminal. In your case, the player keeps drawing until he has more points than the dealer, and you set a flag to influence the next action (-2 pt). This is a severe mistake in an otherwise correct simulator and means that your results in the following two tasks are not interpretable in any way.
Points: 4/6

Task 2

Unrelated to the tasks, but something I have to point out after it was already super annoying in the first task; your loops' exit conditions are very bad Python style. It must be very clear when the criterions are met and looping stops. Shady 'break' statements are not the correct way to do this. For example, in this task, *while game.terminalstate == False:* makes for equivalent and much cleaner code!

The step-size is overcomplicated, but okay this way. Using numpy's *where()* you could have implemented this in a much more efficient vectorized manner. The exploration probability ϵ is correctly calculated. The same is true for the TD-error, and $e(s, a)$ is also nicely incremented, decayed and reset. Overall, *Sarsa(λ)* was flawlessly implemented, but the simulator problems mean that the agent skips too many states to effectively learn. Finally, the experiment over many different λ is correctly designed.
Points: 7/7

Task 3

The *CoarseCoding* class is as unnecessary as it is undocumented. I really have no clue what *phiSet* is supposed to be, but this structure as well as *Qaction()* make no sense at all. The whole point of this approximation is that we do not want to have these tables in memory, using only θ and $\phi(s, a)$ instead (-1 pt). To make this clear: $\phi()$ should not be precomputed,instead we retrieve each $\phi(s, a)$ as (s, a) come up, use it to estimate $Q(s, a)$ and then discard it again. The only thing we save and adapt is θ. This part of the task you execute pretty well. Action selection is sensible, θ is correctly incremented (good eye on the binary features), reset and applied. Overall, the only reason you have weird results is the error in task 1.
Points: 6/7

Total Points: 17/20 Points