Individual Contributions

1. Determining mathematically how the probability function works and explaining in pseudo-code how to code it to Iou-Sheng Cheng and XianXing Jiang. Reviewing code and debugging after it is written.

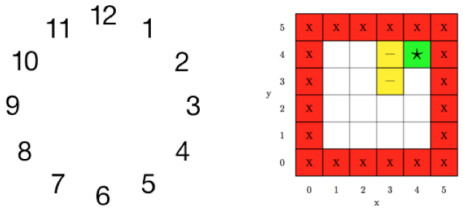

Figure 1: Grid World and robot heading

2. Helping to explain policy iteration, give mathematical equations for correct implementation, and help debug code during its implementation.

$$V^{\pi}(s) = E\Big[\sum_{i=1}^{T} \gamma^{i-1} r_i\Big] \forall s \in \mathbb{S}$$

$$V^*(S) = \max_a \Big[R(s,a) + \gamma \sum_{s' \forall \mathbb{S}} p(s'|s,a) V^*(s')\Big]$$
$$V^*(s) = \max_{\pi} V^{\pi}(s) \ \forall s \in \mathbb{S}$$
$$\pi^* = arg\max_{\pi} V^{\pi}(s) \ \forall s \in \mathbb{S}$$

$$V^*(s) = \max_a Q^*(s,a) \ \forall s \in \mathbb{S}$$

Figure 2: Equations used in Lab

3. Explaining value iteration from a theoretical standpoint to Iou-Sheng Cheng and XianXing Jiang. Helped debug and verify that code was producing the expected results. Reviewed code to check for errors in implementation when it was finished.

4. Wrote report introduction, and helped write overview and sections 2 through 7. Proofread report and helped place in LaTeX using Overleaf.

```
\begin{align}
    V^* (s)     &=
    \underset{a}{\max} \  \Big[ R(s,a) + \gamma \sum\limits_{s' \forall
    \mathbb{S}} p(s'|s,a)V^*(s') \Big]
    \label{eq:8} \\
    V^* (s)     &=
    \underset{\pi}{\max} \  V^\pi (s)\  \forall s \in \mathbb{S}
    \label{eq:9} \\
    \pi^*       &=
    arg \underset{\pi}{\max} \  V^\pi (s)\  \forall s \in \mathbb{S}
    \label{eq:10}
\end{align}

The optimal Q-function \cref{eq:11} will give the total expected reward
gained by the robot given that the robot starts in state s, takes action a,
and then continues to take the best action for the rest of its movement.

\begin{align}
    V^* (s) = \underset{a}{\max} \  Q^* (s,a) \  \forall s \in \mathbb{S}
    \label{eq:11}
\end{align}
```

```
        pe,rows);
    value_difference = sum(sum(abs(value_prev-
        optimal_value_value)));
    optimal_value_value = value_prev;
end
computeTime_valueIteration = toc;

% Display initial state
robot((rows+1)-s(2),s(1)) = s(3);

% Plot robot state
% Grid World has the same configuration as the figure shown in lab3
    instruction
% Value shown on the graph is the heading
% Next action and prerotation is shown in the x axis
% when we reach the goal block, break out of the loop
for j = 1:1000
    act = policy_v((rows+1)-s(2),s(1),s(3),:);
    plot_route(robot,pre,act);
    optimal_policy_value(j,1) = act(1,1,1,1);
    optimal_policy_value(j,2) = act(1,1,1,2);
    if (s(1) == 5 && s(2) ==5)
        break;
    end
    robot((rows+1)-s(2),s(1)) = 0;
    [s pre]= next_state(pe,s,act);
    robot((rows+1)-s(2),s(1)) = s(3);
end
```

Figure 3: Right: LaTeX Code excerpt from final report. Left: Matlab code except from project