**Week 1**:

Consider a software developer working on developing code for an application. While designing the code, the coder is bound to get stuck on a persistent and complex problem for the entire day. In a normal scenario, the developer would get bogged down in designing the code, be it complex or redundant, and hence would require a lot of time and effort in implementing other tasks. Incorporating extreme programming in the development process promulgates the value Courage, enabling the development team to quickly resolve the issue the very next day. In the first week of working on the team project, my team and I were in a dilemma in choosing between Greenfoot and Processing, neither of which we are acquainted with. Though unsure whether the software would be apt for our project, I urged the team to go ahead with Greenfoot. I realized that everyone who is part of the team must be given room and freedom to think about the project, be innovative, and express ideas, no matter how uncertain the final outcome may be.

**Week 2:**

Name: Ashik Panchangam

Pattern: Chain of Responsibility

Value: Courage

GitHub Id: ashikpanchangam

In the second week, my team and I discussed upon creating wireframes for the user interface and the technology for developing our application. Everyone was assigned to work upon different wireframes. Also, everyone was assigned work to design class diagrams for the patterns they are working upon. My team and I are new to Greenfoot. So we decided to learn how Greenfoot functions, by developing simple apps on it. I took the initiative to develop a simple game to understand the functions offered by Greenfoot. Although everybody finished creating class diagrams for their patterns, while other members of my team finished designing wireframes, I was unable to complete the UI wireframe on time. Working on other projects became an impediment to developing the wireframes on time. Communication within the team was bad as a few members within the team weren’t notified about the changes and commits made on GitHub. One of my team members came up with an idea to evade confusion and promote good communication, by suggesting on maintaining journals on Git repository. I urged the other team members to implement this idea in order to maintain good communication within the team.

**Week 3:**

Name: Ashik Panchangam

Pattern: Chain of Responsibility

Value: Courage

GitHub Id: ashikpanchangam

In the third week, my team and I discussed upon integrating the patterns for developing our code. I realized the need to continuously integrate patterns into the code that was being developed. Although the team was unsure if it would create a potential hassle, I urged the team to continuously integrate the code of the patterns that is being developed. The application we are working on has 3 observers that require to be notified based on the subject's logic. A single team member working on implementing the observer pattern logic for the entire code was rhetoric and time consuming. To improve the efficiency and time management, I suggested on two members working simultaneously on developing the code using observer patterns. The entire team is unsure if such an implementation would create conflicts while integrating the code later, but, it has been recognized as the only way to improve efficiency for working together as a team. There wasn't good clarity about the factory and builder patterns, both of which seemed to be apt for nudging into our application. Although uncertain, the team decided to implement builder pattern in the application. Eventually, the implementation turned out to be wrong, and hence, the more sensible pattern, "The factory pattern" is being used in our application.

**Week 4:**

Name: Ashik Panchangam

Pattern: Chain of Responsibility

Value: Courage

GitHub Id: ashikpanchangam

In the fourth week, my team and I worked upon integrating the code. While integrating the code we faced difficulty in getting the Car objects to face in the right direction. Later, the issue was resolved after understanding the functions offered by Greenfoot. I created test cases for a certain user interface. Creating the use cases was quite challenging to me as it required concrete understanding of the game from a technical and user perspective. That being said, I studied and envisaged a few real life scenarios to create a few use case scenarios and a Use case diagram to represent our application. Currently, the application has just one level. I am urging the team to add more levels to the game to increase its difficulty and make it more alluring. That being said, the application development is nearly done, with a few changes needing to be made to the user interface. In a nutshell, I would say taking up the challenge of developing a very interesting game of chance has been exciting. We as a team, had to come together to work effectively. Moreover, I personally faced a lot of challenges while incorporating the “Chain Of Responsibility” pattern in application. I had to learn using Greenfoot and understand in tandem the pattern incorporated by my other teammates. In addition, developing test cases and use cases for more than one scenario was challenging. Sometimes different work assigned to different teammates couldn’t be completed on time. The audacity to learn new methods to develop certain components for our application has been the notable aspect of us coming as a team and working together. Currently, the user interface requires some changes to be made on it, which should be done by the mid of this week.