Mountaineering
Final Project Summary
Group 5 - Zheng Fang, Ke Sun, Prathyusha Yelleni

This project implemented a cool board and card game with a domain in sport mountain climbing with critical-thinking and strategic thinking. The goal of this project to create a strategic and fun game for players of any ages. By adding the theme of mountain climbing, we can teach players the dangers and benefits of mountain climbing as a sport or possibly motivate players into doing mountain climbing.

The game will be developed with single player in mind first. The game will start with each player and computer AI starting with five cards. Players begin their turn by drawing two cards. The cards are represented using regular card standards (Ace, 2-10, Jack, Queen, and King). The cards that range from 2-9 will be movement cards. These determine the number of spaces you can move. You are only allowed to move once per turn and only forward. The player can also use special ability cards. The goal of the game is to reach the top of the mountain before everyone else.

Requirement is also elicited. The game will be launched once the .exe file is double clicked. The game will load the Main Menu. In the menu the player can either choose to create a new user name. After entering into the game, a world map will be displayed. The player can choose a mountain on the map and click to climb the mountain. When a mountain is chosen and clicked, the mountain map will be loaded. The goal of the player is to reach the top of mountain earlier than the computer AI from the same starting point. A deck of cards will be displayed at the middle of the screen and they will be shuffled automatically by the game. The player and AI will be distributed with 3 cards each, and then play the card game.

Player should feel consistency during the entire play of the game. The game should not slow down periodically. Any scenario change should be finished in 0.5 seconds. Loading of animation and cartoon should be completed in 2 seconds. Also, the game should automatically save current progress in 5 minutes into a temporary file. If an unintentional shutdown occurs, the player should be able to recover most of the game. Additionally, the game should simulate the real mountain environment. The game is sold as a single CD with proper packing in most regions; usually one copy could be installed on 3 different computers.

In the system design section, Sequence diagrams were created to illustrate the interaction between players and system. The series of actions were depicted to respond player action. We created eight subsequence diagrams which include Save Game, Create Game, Load Game, Card Distribution, Store Purchase, Play Card, Hit a Block and Check Personal Belongings.

Design goal was also stated which includes a performance criterion, a dependability criterion, a maintenance criterion, a cost criterion and end user criterion.

The interface will be written based on different subsystems. We can focus on response time as the system is not a complex one. We can put more time on coding to implement some algorithms that has a smaller time complexity. Additionaly, we will use class diagram to interpret each subsystem and classes in each subsystem. Each subsystem will be incorporated into a package. And each package have appropriate interface to communicate with other packages. Then we listed all class diagrams towards object design, which will be categorized as to each subsystem.

Acceptance test is concisely stated in this chapter. Since small team working on this project, therefore, periodical goal is set separately, and a couple of releases will be illustrated as well. And the project is decomposed into 2 releases to better manage the project process for small development team.

This semester, we have been working on the monoplay like game mountain climbing. Through development of this game, we have learned how to complete requirement elicitation, analysis, system design, object design and acceptance test requirement. In this development process, we did learn some methods for Object Oriented Software Engineering, such as how to implement UML diagrams for internal design.