

Name: Yuhan Wu

ID: 21064228

Project: RAllnet

Breakdown of the project:

1. Ability and its super classes:

- Base class: Ability (abstract)
- Super classes:
 - o abilityL
 - o abilityF
 - o abilityD
 - o abilityP
 - o abilityS

2. Link class:

- define the link objects, their strength, type, and the player they belong to.

3. Player class:

- owns an array of links

4. Board class:

- The core class in the game which coordinates the interactions between player, exhibits the effects of each move, defines the overall game logic and functionalities required.

5. The Observer Pattern:

- Abstract observer class: observer
- Abstract subject class: subject
- Concrete subject class: Board
- Concrete observer classes:

TextObserver

graphicalObserver

1. implement Core classes: Nov. 20th
 - a. Board
 - b. Link
 - c. Player

2. Implement the Ability class and its super class: Nov. 23th
 - a. Base Class
 - b. Super Class
 - c. Impacts on the core performances

3. Implement win/lose condition: Nov. 24st
 - a. Win
 - b. Lose

4. Implement observer class and ability objects: Nov. 24st
 - a. Observer class pattern
 - b. Ability objects

5. Implement the logic between two players: Nov. 25nd
 - a. When does each player take the turn
 - b. Input handling

6. Implement the textObserver: Nov. 25rd
 - a. Printing to the output stream
 - b. Updating scores and grids

- | | |
|--|-----------------------|
| 7. Implement the graphicalObserver: | Nov. 25 th |
| a. Present graphics to the output stream | |
| | |
| 8. Overall game flow: | Nov. 26 th |
| a. Debug and make sure the overall game flow is correct. | |
| | |
| 9. Implement the extra functionalities: | Nov. 26 th |
| a. Implement the extra 4 abilities required | |
| b. 4-player game mode added | |

Question1 : In this project, we ask you to implement a single display that maintains the same point of view as turns switch between player 1 and player 2. How would you change your code to instead have two displays, one of which is from player 1's point of view, and the other of which is player 2's point of view?

Solution:

I am using the observer pattern to print to the output.

For my graphical and text observers, there will be a field indicating which player's view is being presented. In the notify method, representations will be designed in different ways depending on the specific player.

Question 2 : How can you design a general framework that makes adding abilities easy? To demonstrate your thought process for this question, you are required to add in three new abilities to your final game. One of these may be similar to one of the already-present abilities, and the other two should introduce some novel, challenging mechanic. You are required to actually implement these.

Solution:

I plan to use the base and super class pattern to achieve this. I will design the abstract base class as class Ability, with several virtual methods like use() which applies the effect of this

ability. I will design each of the specific abilities to be the super class of the class Ability, and they will override the public methods to correctly apply their own effect to the game.

In this way, if I want to add more abilities, I only need to declare it as a super class of the abstract base class Ability, and override those performance methods.

Question3: One could conceivably extend the game of RAllnet to be a four-player game by making the board a “plus” (

+

) shape (formed by the union of two 10x8 rectangles) and allowing links to escape off the edge belonging to the opponent directly adjacent to them. Upon being eliminated by downloading four viruses, all links and firewall controlled by that player would be removed, and their server ports would become normal squares. What changes could you make in your code to handle this change to the game? Would it be possible to have this mode in addition to the two-player mode with minimal additional work

Solution:

In my board class, I have integer fields: max_row, max_col, to indicate the size of the board; I have integer field num_player to indicate how many players we have in the game. The printing behavior will depend on the size of the board and the number of players in the game.

In order to achieve the effect described, we can reset the dimensions of the board, and update the num_player field.

To let links to escape off the edge belonging to the opponent directly adjacent to them, we can modify the move method in Board class to make it behave differently depending on the number of players in the game.

To achieve eliminating all links and firewall controlled by that player, we can update the observer pattern so that the links and firewalls in the Board class are automatically updated if any player exits the game. This functionality will be activated by the num_player == 4. So that the game will switch from 2-player-mode the 4-player-mode

By implementing those extra features without modifying the design for 2 players' game design, we can switch between those two modes depending on the number of players we want in the game.